AD-A080 419 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 1/3
THE APPLICATION OF OUTPUT PREDICTIVE DIGITAL CONTROL TO WING FL--ETC(U)
DEC 79 D E CHAFFIN
UNCLASSIFIED AFIT/GE/EE/79-9 NL

1 OF 2
AD A080419

AFIT/GE/EE/79-9

(1)

LEVEL II

(6) THE APPLICATION OF OUTPUT PREDICTIVE DIGITAL CONTROL TO WING FLUTTER SUPPRESSION AND TERRAIN FOLLOWING PROBLEMS

(9) Master's THESIS

(14) AFIT/GE/EE/79-9

(12) David E. Chaffin
Captain USAF

(11) Dec '79

(12) 177

Approved for public release; distribution unlimited

AFIT/GE/EE/79-9

THE APPLICATION OF OUTPUT PREDICTIVE DIGITAL CONTROL TO WING FLUTTER SUPPRESSION AND TERRAIN FOLLOWING PROBLEMS

THESIS

Presented to the Faculty of the School of Engineering

of the Air Force Institute of Technology

Air Training Command

in Partial Fulfillment of the

Requirements for the Degree of

Master of Science

by

David E. Chaffin

Captain USAF

Graduate Electrical Engineering

December 1979

Preface

This work is basically a continuation of Captain Howard Colson's thesis study (Ref 1). Output Predictive Control is a relatively new concept and little material has been published on the subject.

Guidance and clarification throughout the project was from Major J. G. Reid, my thesis advisor. Many thanks to him for his patience and indulgence. I am also obliged to Captains Silverthorn and Rader for their assistance during the research and in the preparation of the final document.

Thanks especially to Risa, Damien and Sally for their toleration during this AFIT tour.

David E. Chaffin

Contents

List of Figures

List of Tables

Abstract

This thesis is a study of a digital control technique known as Output Predictive Control (OPC) or Model Algorithmic Control (MAC). In OPC, the behavior of the system is predicted using its impulse response function and the desired response is characterized by a reference trajectory. Controls are computed iteratively to drive the system output along the desired trajectory.

In an earlier study, the system was made to follow the reference trajectory exactly, but only at the control application time; there were large oscillations of the output between control changes. In this study, the control calculation is reformulated as a least squares curve fit problem, allowing some deviation from the desired trajectory.

This approach is applied as a regulator for a very lightly damped fourth-order single-input/single-output system and as a pitch axis autopilot in a simplified terrain following problem. A qualitative discussion of robustness properties is included.

The design of the controller is difficult due to the interrelationships of the internal parameters; however, the results of the terrain following example indicate that this is a viable approach for this problem.

THE APPLICATION OF OUTPUT PREDICTIVE DIGITAL CONTROL TO WING FLUTTER SUPPRESSION AND TERRAIN FOLLOWING PROBLEMS

I Introduction

Background

A new control technique known as Output Predictive Control (OPC) or Model Algorithmic Control (MAC) has recently been developed (Ref 6,7, 8,11,12). The control technique differs from previous state variable methods in that it employs a discrete impulse response model rather than a "state space model." Also, rather than employing "feedback," OPC uses an explicit prediction of the future response, thus trying to find the future input which best matches a desired future response.

This technique is conceptually pleasing in that it approximates a human "controller" in some tasks. As an example, consider a pilot attempting to maintain a particular course. This course can be referred to as a 'set path' and the control objective is to maintain the set path. If an error exists between the aircraft's actual course and the set path, the pilot formulates a control input (or series of control inputs) to return the aircraft to the set path. Some of the factors involved in his control calculation include:

1. The zero-input response of his aircraft, i.e., the path the aircraft will follow if no additional controls are applied.

2. The urgency of the error condition, i.e., how quickly must he get back to the set path? The pilot will want the error reduced rapidly if maintaining his present (incorrect) course leads to a hazardous situation (i.e., collision with a mountain or another aircraft).

3. Aircraft limitations, his desire for a smooth maneuver or passenger comfort, and the urgency of the situation will determine the trajectory to fly from his present position to a point on the set path. The most direct approach is to fly a heading perpendicular to the set path. This would, however, result in passing through the set path and an even larger deviation or overshoot on the other side. A more practical solution is to choose a trajectory which will reduce the course error more gradually and allow the pilot to roll out on the set path.

4. The "system model" of the aircraft (i.e., how the aircraft will react to a given control input). Through experience, the pilot knows his aircraft's response to a control change at a particular altitude, airspeed and aircraft configuration.

Based on these factors, the pilot formulates a control scheme to fly his aircraft to the set path. After the control has taken effect, he assesses the results and, if necessary, reformulates the problem. During the landing phase, when course (and glide path) control is critical, the entire sequence can occur in less than a second.

OPC or MAC is an attempt to automate this type of process using a digital computer. The discrete impulse response model is used both to make an explicit prediction of future output responses using knowledge of past inputs and to compute alternative future control strategies. The one control strategy is then selected which gives the best match to

a desired future output trajectory. The output prediction and control computation are performed closed loop on a discrete sample by discrete sample basis.

Original applications included industrial process control where a priori models were not well understood (Ref 12). In these situations, on line identification of the discrete impulse response function model can be employed.

The control strategy has also been successfully applied to a number of aerospace problems (Ref 6,8) and theoretical investigations (Ref 7). In many of these aerospace applications, the system is very lightly damped, some nonminimum phase. In such situations, it was found that a direct application of the output predictive control strategy to force the output to follow the desired path exactly would lead to closed loop instability almost immediately (Ref 1). Specifically, it was found that the system could be made to follow the desired trajectory exactly, but only at the discrete sample times corresponding to a change in control. The response could be rapidly oscillating between control changes, leading to instability (Ref 1).

Problem and Scope

The objective of this thesis effort is to formulate and test a control strategy which will not only follow the desired trajectory at control change times, but also keep output oscillations within "acceptable" bounds between control changes.

Single-input/single-output (SISO) reduced order system models will be used to investigate the control technique for use as a regulator for a very lightly damped system and as a pitch controller for a fighter-type aircraft.

This report will not go deeply into the background theory of OPC; the interested reader is referred to References 6, 7, 8 and 12. This report will focus on the controller used to achieve the objective as listed above.

A complete computer program called IDCOM (Identification and Command) has been designed by the French company of Andersa/Gerbios to implement their version of the output predictive technique. IDCOM is not available for proprietary reasons, but the controller implemented in this thesis is conceptually similar.

Approach and Sequence of Presentation

The solution proposed in this report is to formulate the prediction/control computation so that the future output trajectory is calculated at a smaller sample spacing than the control switch times. Trying to match the desired path at this finer spacing then creates an overdetermined, weighted least squares problem and allows prediction times into the future as long as are computationally feasible. The internal energy states (causing oscillations between control changes) are thus controlled indirectly by keeping the future output trajectory within a least squares "tube" rather than attempting to match the desired path exactly at only control switch times.

The theory and equations used in this scheme are presented in Chapter II. Internal parameters and the effect of their variations on the sampled output and control energy expended are discussed in Chapter III. A qualitative discussion of "robustness" in the presence of model mismatch is presented in Chapter IV. Application of the control scheme as a pitch controller for a reduced order model of a fighter aircraft is presented in Chapter V. Finally, conclusions and recommendations for further study are listed in Chapter VI.

II Theory

This chapter outlines the theory and equations underlying the output predictive controller that is developed. This development will be for the algorithm applied as a regulator. Modifications for application of the algorithm to the aircraft pitch controller and terrain following problems are discussed in Chapter V.

Figure 1 is a functional block diagram of the controller algorithm. In this formulation of OPC, a desired trajectory (y_d) is calculated from the point of the 'present' system output (y) to the set point (y_{set}) . A control input is calculated to zero out the difference (z) between the system's zero-input response (y_{zi}) and the desired trajectory (y_d) . The following sections explain each of the functional blocks in detail.

Discretization of System Model

This thesis treats the discrete time control of the nth order linear time invariant, single-input system represented as

$$\dot{\underline{x}}(t) = A\underline{x}(t) + Bu(t) \tag{1}$$

$$\underline{x}(0) = \underline{x}_0 \tag{2}$$

with single-output sampled at a constant sample time, T ,

$$y(kT) = C\underline{x}(kT) \tag{3}$$

Assuming a piecewise constant input, the discrete time model of the system (1) - (3) is then represented by

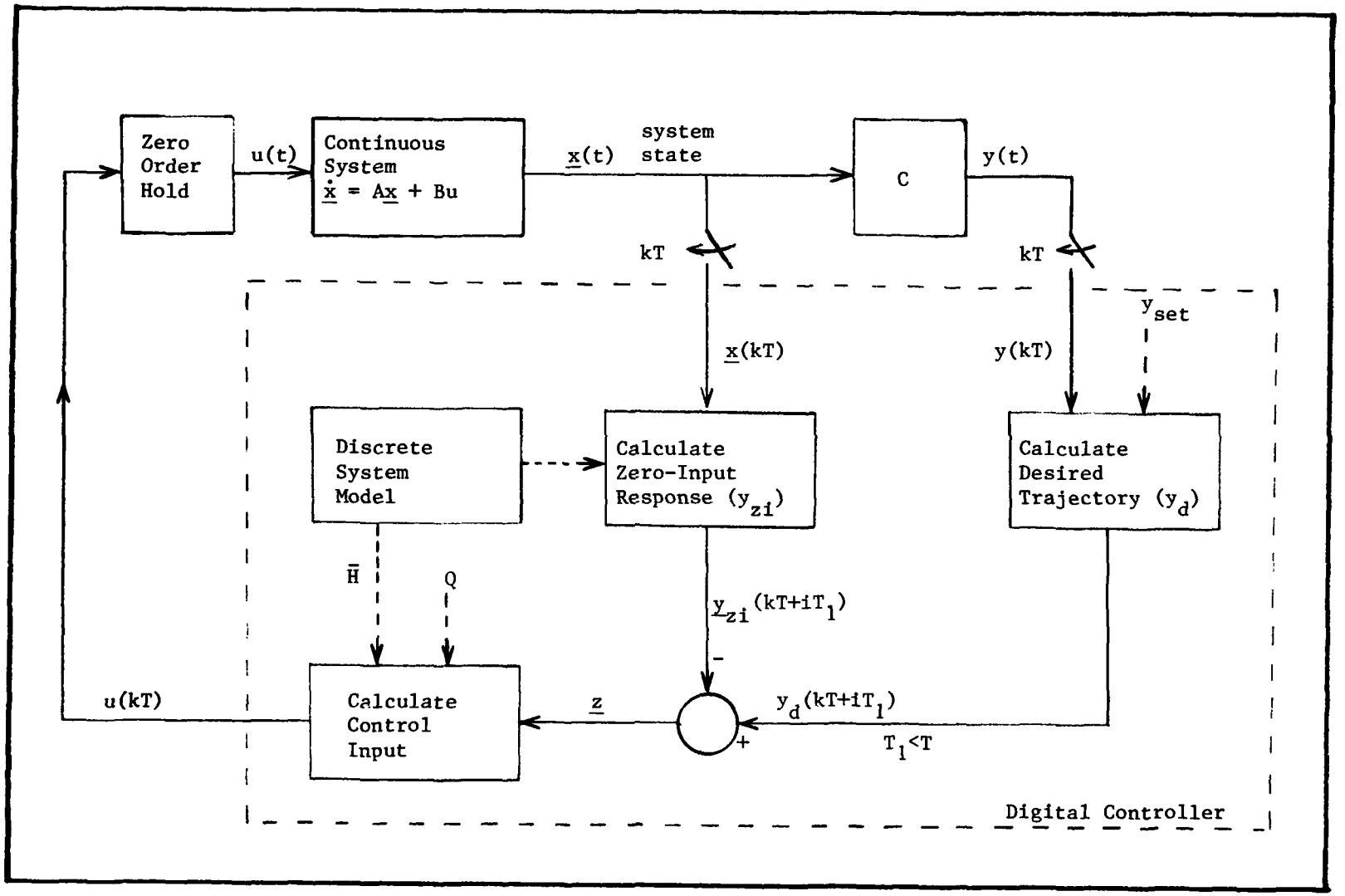


Fig 1. Functional Block Diagram of the Output Predictive Controller

$$\underline{x}(k+1) = F\underline{x}(k) + Gu(k) \tag{4}$$

$$\underline{x}(0) = \underline{x}_0 \tag{5}$$

$$y(k) = C\underline{x}(k) \tag{6}$$

where

$$F \equiv e^{AT} \tag{7}$$

$$G \equiv \left(\int_0^T e^{A\tau} d\tau\right)B \tag{8}$$

The system (1) - (3) then has the discrete impulse response sequence (Ref 10,11) denoted as

$$\{h(1),h(2),h(3), \ldots\} = \{CG,CFG,CF^2G, \ldots\} \tag{9}$$

or

$$h(i) = CF^{i-1}G \tag{10}$$

Control Objectives

The controller is first implemented as a regulator. As pictured in Figure 2, the system output is taken from an arbitrary initial point (state) along a reference output trajectory to a desired final value or set point. As an example, consider the pointing of a radar antenna aboard a ship rolling and tossing in rough water. If the desired antenna angle is constant at θ_x (referenced to true north or some inertial reference) the set point for the regulator is θ_x . The controller's task then is to drive the antenna to, and keep it at, θ_x , regardless of the gyrations of the ship.

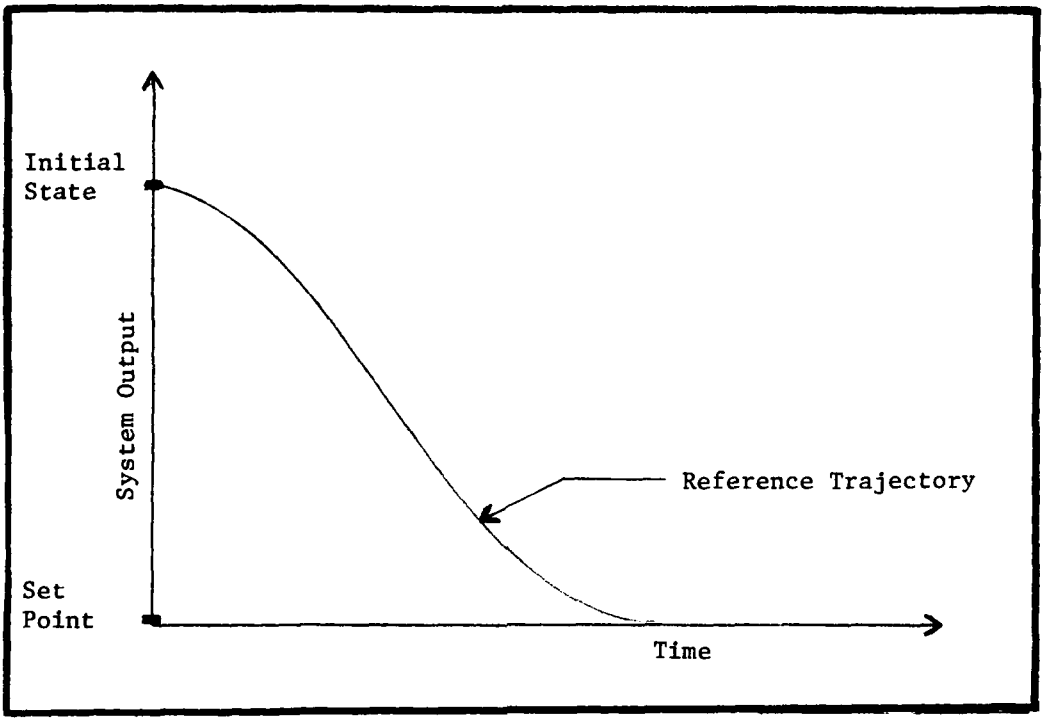


Fig 2. Controller Implemented as a Regulator

In the tracking application discussed in Chapter V, the controller is used as a pitch axis autopilot. Instead of a constant set point, the system is driven to a time varying commanded pitch angle as dictated by the requirement to fly along a given terrain profile.

Calculate Points Along
Desired Trajectory

In the regulator implementation of the proposed controller, the reference or desired trajectory is chosen as a decreasing exponential with time constant Tau. Discrete points along the reference trajectory shown in Figure 2 can be calculated as

$$y_d(i) = y_{set} - \phi^i(y_{set}-y) \qquad (11)$$

where

$y_d(i)$ = discrete point along desired path

y_{set} = set point or where we want the system to go (for the regulator, $y_{set}=0$)

y = 'present' output of the system

ϕ = exp ($-T_1/Tau$)

T_1 = sample time at which $y_d(i)$ is calculated

Tau = time constant of first order decreasing exponential representing reference trajectory from y to y_{set}

Figure 2 illustrates the situation for initial start-up of the system. For each subsequent iteration of the digital algorithm the "present" system output is used for the initial state and a new reference trajectory is calculated for each iteration.

Zero-Input Response

The total system response can be represented as

$$y(t) = y_{zs}(t) + y_{zi}(t) \qquad (12)$$

where

$y_{zs}(t)$ = zero-state response

$y_{zi}(t)$ = zero-input response

If the system is started at rest (i.e., zero internal energy), the response to an input would be termed the system's zero-state response.

The system response at time k based solely on past inputs (i.e., all future inputs equal zero), is the zero-input response. The

zero-input response may be found in terms of the impulse response function (Ref 10,11) or in terms of the discrete system model equations as

$$y_{zi}(i) = CF^{i}\underline{x}(0) \tag{13}$$

The algorithm is an iterative technique where the "present" time is always considered to be $t=0$. When the program is first started $\underline{x}(0)$ is the arbitrary initial state of the system. For each successive iteration $\underline{x}(0)$ is the actual system state, thereby closing the loop. It is beyond the scope of this thesis to consider the "state estimation problem," but clearly a closed loop estimator of the "state" is required for this particular implementation of the prediction calculation.

Control Calculation

It is desired that the system output match some reference path, or in equation form

$$y_d(i) = y(i) \quad ; \quad i = 1,2,..,N \tag{14}$$

where

$y_d(i)$ = points along desired path

$y(i)$ = system output

The discrete form of Eq (12) is substituted for the right side of Eq (14):

$$y_d(i) = y_{zi}(i) + y_{zs}(i) \tag{15}$$

An expression relating $y_{zi}(i)$, the zero-input part of the response, to the current system "state" was given by Eq (13).

Using the ideas of a piecewise constant input and a discrete representation of the convolution integral (Ref 10) gives an expression for the zero-state response:

$$y_{zs}(i) = \sum_{j=1}^{i} h(j)u(i-j) \tag{16}$$

Substituting Eq (16) into Eq (15) and expanding yields

$$\begin{bmatrix} y_d(1) \\ y_d(2) \\ y_d(3) \\ \vdots \\ y_d(N) \end{bmatrix} = \begin{bmatrix} y_{zi}(1) \\ y_{zi}(2) \\ y_{zi}(3) \\ \vdots \\ y_{zi}(N) \end{bmatrix} + \begin{bmatrix} h(1)u(0) \\ h(2)u(0)+h(1)u(1) \\ h(3)u(0)+h(2)u(1)+h(1)u(2) \\ \vdots \\ \sum_{j=1}^{N} h(j)u(N-j) \end{bmatrix} \tag{17}$$

This equation is solved, with no restrictions, in Ref 1 for the input as

$$u(0) = \frac{y_d(1) - y_{zi}(1)}{h(1)} \tag{18}$$

This is, in effect, one step ahead prediction and led to instability.

If the problem is restricted to bring in the idea of "smoothing terms," or equivalently, if the input is held constant over NSM (number of smoothing terms) of the desired points ($y_d(i)$), an overdetermined least squares problem evolves. Figure 3 illustrates the concept of four smoothing terms within the span of one control input.

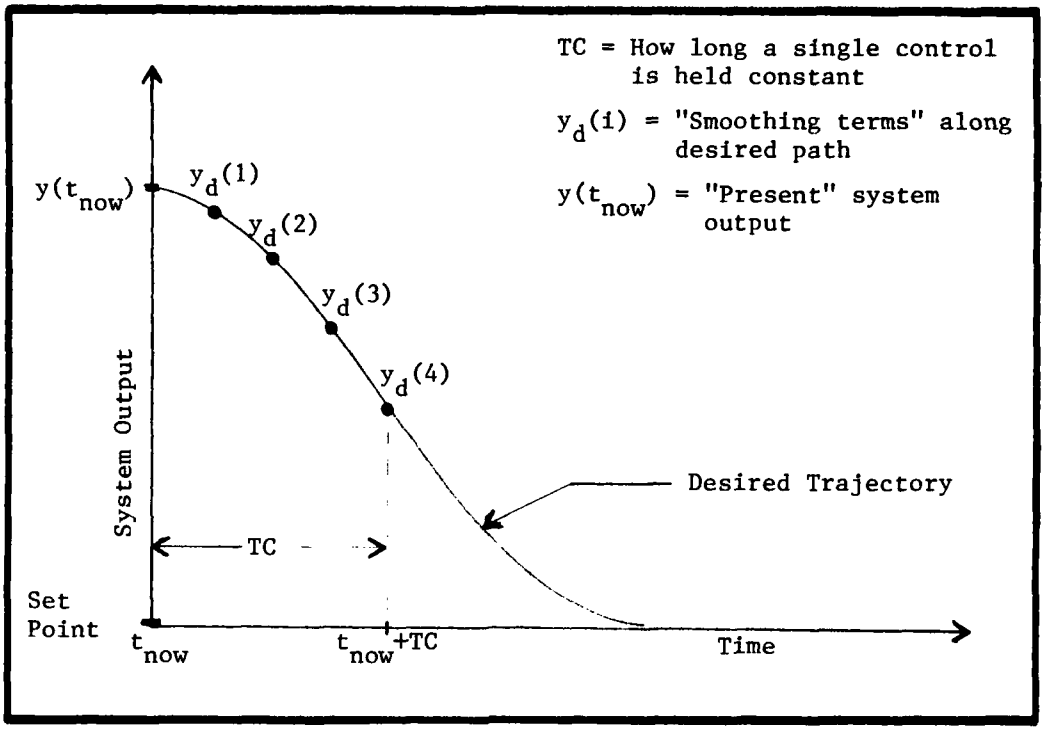


Fig 3. Four Smoothing Terms Within One Control Span (NSM=4)

A new set of controls can be defined as

$$\begin{aligned}
&[\quad u(0),u(1),..,\ u(NSM)\quad] = \bar{u}(0)\\
&[u(NSM+1),u(NSM+2),..,\ u(2*NSM)] = \bar{u}(1)\\
&\qquad\qquad\vdots\qquad\qquad\qquad\vdots\\
&[\ .\ .\ .\qquad\qquad ,\ u(N)\quad] = \bar{u}(L)
\end{aligned} \tag{19}$$

where $\bar{u}(0)$ is not a row vector, but the value assigned to

u(0),u(1),...u(NSM).

Eq (19) can be substituted into Eq (17), then manipulated to yield

$$\underline{z} = \bar{H}\underline{\bar{u}} \tag{20}$$

$$\underline{\bar{u}} = \begin{bmatrix} \bar{u}(0) \\ \bar{u}(1) \\ \vdots \\ \bar{u}(L) \end{bmatrix} \tag{23}$$

Points along the desired trajectory, $y_d(i)$, and the zero-input response are still given by Eqs (11) and (13) respectively. $\underline{\bar{u}}$ is a vector of inputs with dimension "L," the number of future inputs to be calculated. $\bar{H}$ is a matrix of impulse response functions, similar in information content to the Hankel matrix (Ref 11).

Eq (20) is a linear equation which can be solved for $\underline{\bar{u}}$. If the control problem is reformulated using the smoothing terms approach, the result is an overdetermined problem; the "best" solution for $\underline{\bar{u}}$ is a weighted least squares "curve fit." The cost function can be defined as

$$J = (\underline{z}-\bar{H}\underline{\bar{u}})'Q(\underline{z}-\bar{H}\underline{\bar{u}}) \tag{24}$$

where Q is a positive, semi-definite weighting matrix.
For computational simplicity, one can also choose for it to be diagonal. The influence of Q on the controller performance will be discussed in detail in Chapter III.

One way to solve the least squares approximation is to develop the Normal Equation (Ref 5:122-129; Ref 9:72,222-224). Taking the partial derivative of J with respect to $\underline{\bar{u}}$, setting the result equal to zero and solving for $\underline{\bar{u}}$ will yield the Normal Equation:

$$\frac{\partial J}{\partial \underline{\bar{u}}} = -2(\underline{z}-\bar{H}\underline{\bar{u}})'Q\bar{H} = 0 \tag{25}$$

Taking the indicated transpose and solving the equation for $\underline{\bar{u}}$ yields

$$\underline{z}'Q\bar{H} = \underline{\bar{u}}'\bar{H}'Q\bar{H} \tag{26}$$

$$\underline{\bar{u}}' = \underline{z}'Q\bar{H}(\bar{H}'Q\bar{H})^{-1} \tag{27}$$

or

$$\underline{\bar{u}} = (\bar{H}'Q\bar{H})^{-1}\bar{H}'Q\underline{z} \tag{28}$$

The dimensions of the elements of Eq (28) are

$\underline{\bar{u}}$ = An L dimension vector

$\bar{H}$ = An (NSM*L)XL matrix

Q = An (NSM*L)X(NSM*L) matrix

$\underline{z}$ = An (NSM*L) dimension vector

Due to the fact that Q is a diagonal matrix and all of the real information contained in $\bar{H}$ is found in the first column, a memory saving FORTRAN coding technique was proposed by Dr. J. G. Reid and implemented by the author. This technique is illustrated by the solution of the normal equation for a simple example problem in Appendix A.

Eq (28) is used to solve for a vector of inputs $\underline{\bar{u}}$. In terms of on-line implementation, only the first element of $\underline{\bar{u}}$ is used. The problem is reformulated with each iteration and a new $\underline{\bar{u}}$ vector found.

Application of Control

After the control has been found, the system state is updated:

$$\underline{x}(k+1) = F\underline{x}(k) + G\bar{u}(0) \tag{29}$$

$\bar{u}(0)$ is the first element of the vector of controls found using Eq (28). The output of the system can now be found as

$$y(k) = C\underline{x}(k) \tag{30}$$

Summary

The general sample times and index of Figure 1 can now be defined. The overall sample time, T , is the control change time, TC . The system model is discretized at a sample time T_1=TC/NSM . The desired trajectory (y_d) and zero input response function (y_{zi}) are predicted i=1,2,...NSM*L discrete points into the future.

The relationships developed in this chapter have been coded in FORTRAN. Appendix B contains a sample program. Embodied in the program listed are the options of adding zero-mean white Gaussian noise to the sampled input and/or control calculated. Model mismatch options are also provided. These features are discussed in Chapter IV.

A very lightly damped fourth order system (see Chapter III) is chosen as a test model for the regulator and the results of variations of the internal parameters used by the controller

TC = Time span of one control input

Tau = Time constant of reference trajectory

L = Number of future inputs calculated per iteration

Q = Weighting matrix

NSM = Number of smoothing terms calculated per control

NSM*L = Total number of output points predicted into the future

are recorded. The effect on the system output and control energy

required to drive the system to a zero set point for various combinations of internal parameters are the subjects of Chapter III.

III Internal Parameter Variation and Results

The hypothetical regulator described in Chapter II is applied to a lightly damped fourth order system. Throughout this chapter, the control objective is to take the system output from an arbitrary initial state, along an exponential path, to a zero set point. The results of variations of the internal parameters Tau, Q , NSM , L and TC were graphically recorded. Only a few of the infinite number of possible combinations are presented. Because of the interdependence of the parameters, only a general idea of the effects of variations of a particular parameter is possible.

An arbitrary initial state and C matrix are chosen as

$$\underline{x}_0' = [\ 20 \quad 10 \quad 5 \quad 0\ ] \tag{31}$$

$$C = [\ 1 \quad 0 \quad 0 \quad 0\ ] \tag{32}$$

for demonstration of the characteristics of the controller. A random number generator was used at one time to generate an initial state; however, the general characteristics exhibited by the controller were unchanged.

System Model

An all pole fourth order system, described by the differential equation

$$\dddot{y}(t) + 1.6\,\ddot{y}(t) + 274\ddot{y}(t) + 279\dot{y}(t) + 8612y(t) = u(t) \qquad (33)$$

is chosen as the system model. $y(t)$ is defined as an arbitrary output and $u(t)$ as an arbitrary input. The eigenvalues

$$\lambda_{1,2} = \sigma \pm j\omega_d = -0.25 \pm j15.4 \qquad (34)$$

$$\lambda_{3,4} = \sigma \pm j\omega_d = -0.55 \pm j6.0 \qquad (35)$$

are representative of the first and fifth aeroelastic modes of the B-52 Control Configured Vehicle airspeed root locus (Ref 1:8,18; Ref 2:2-20). Figure 4 shows the fourth order system's open loop time response for a unit step input.

Variation of Time Constant of the Exponential Reference Trajectory - Tau

In the regulator application, an exponential path is chosen as the reference trajectory or desired path to the set point. Tau, the time constant of the exponential path, is a measure of how quickly the system output is driven to the set point.

Figure 5 shows the system output for Tau = 0.1 and Tau = 0.3 . Predictably, as the system output is brought to the set point more rapidly, the magnitude of the controls required increases. This is shown in Figures 6 and 7.

Variations in the Weighting Matrix -- Q

The Q matrix in Eq (28) is defined as a positive semi-definite, diagonal weighting matrix. In the overdetermined least squares problem

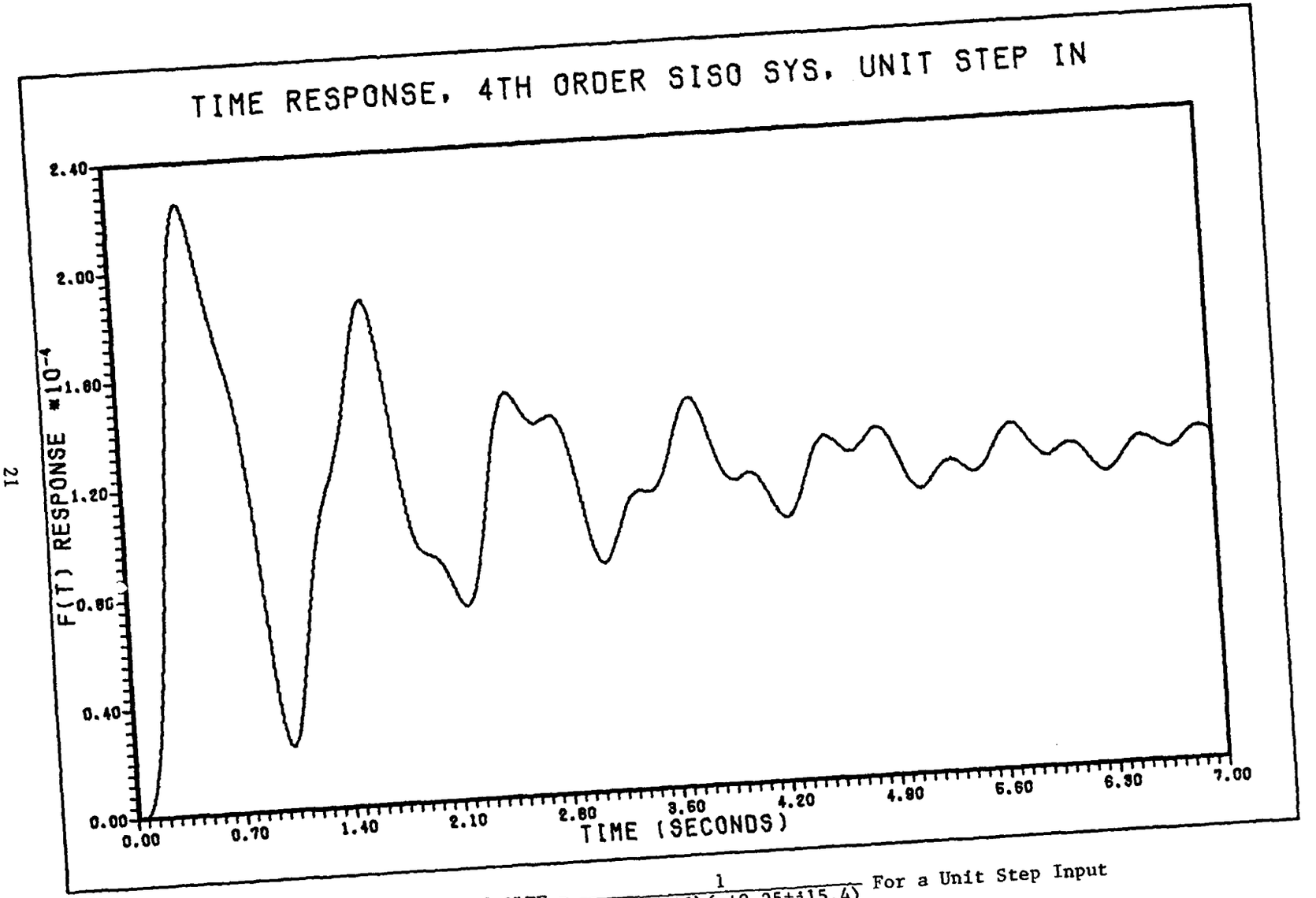


Fig 4. Time Response of OLTF = $\frac{1}{(s+0.55\pm j6)(s+0.25\pm j15.4)}$ For a Unit Step Input

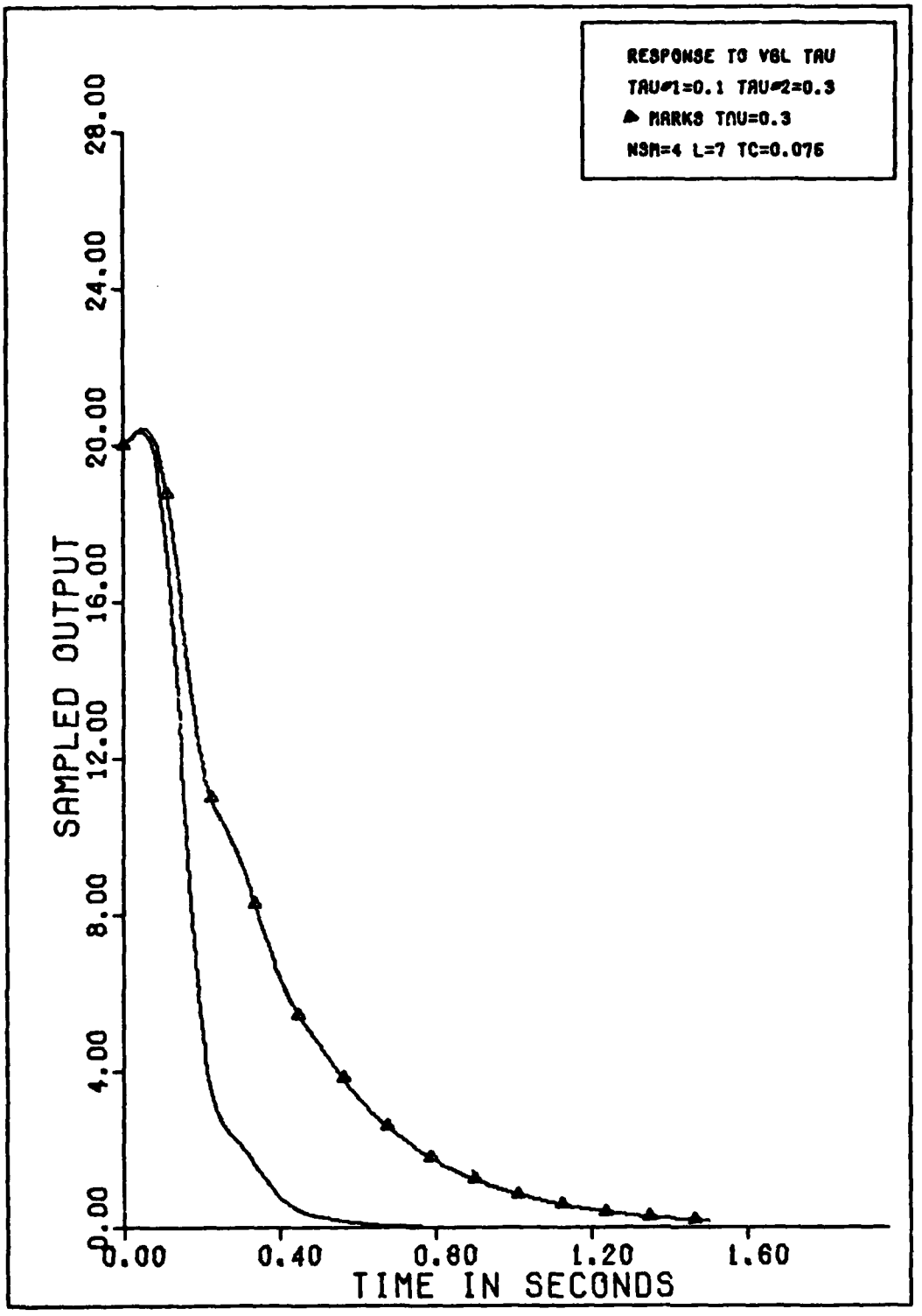


Fig 5. System Output for Tau=0.1 and Tau=0.3 (Δ)

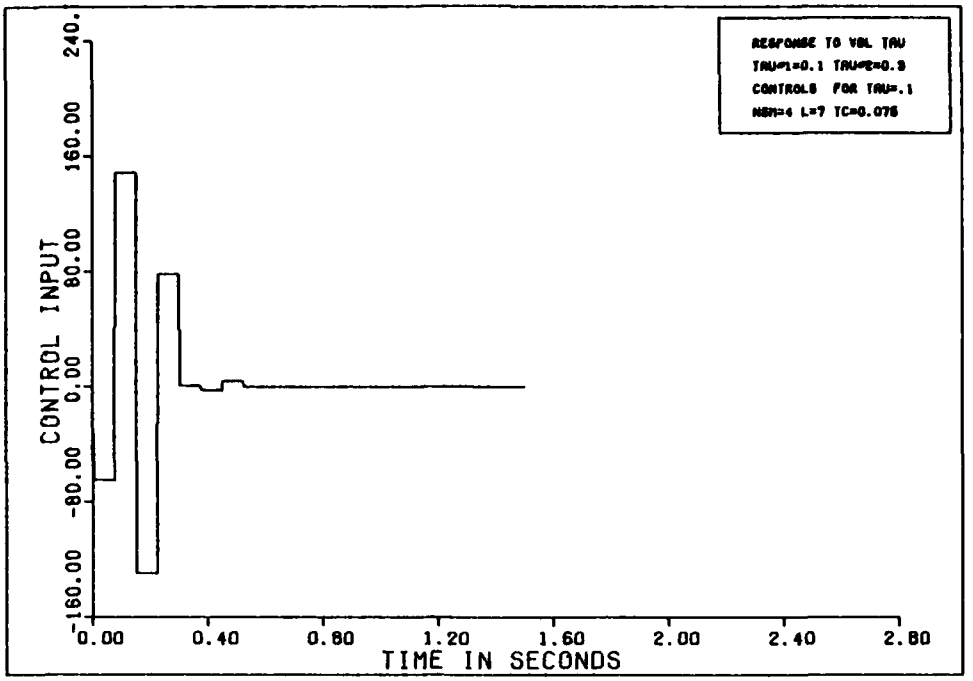


Fig 6. Controls Applied for Tau=0.1

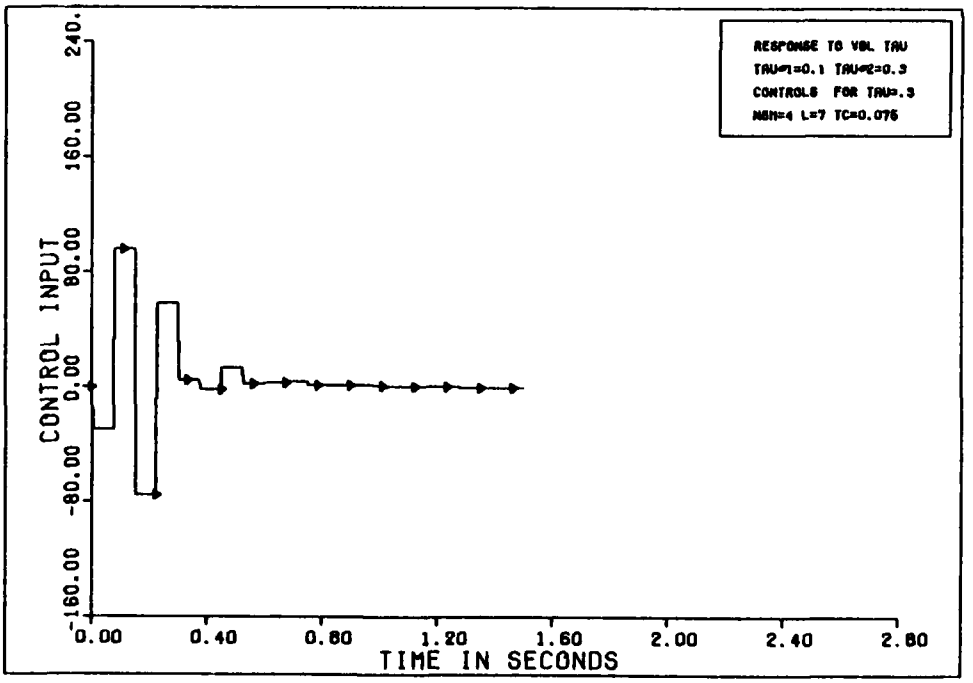
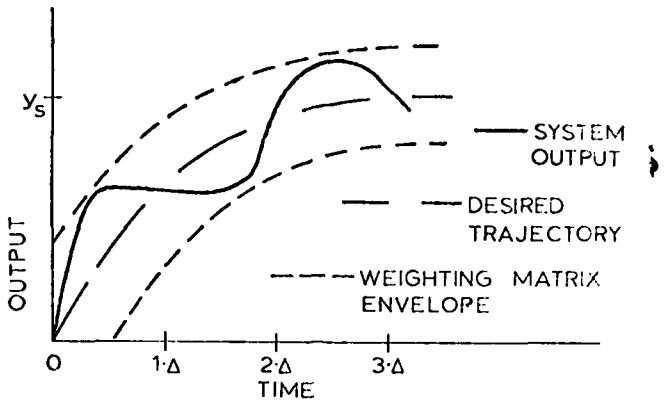
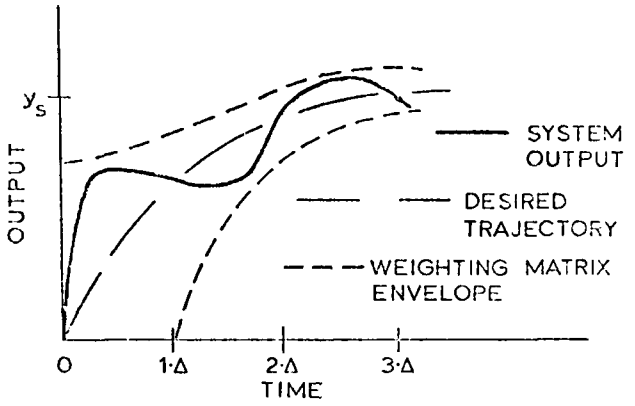


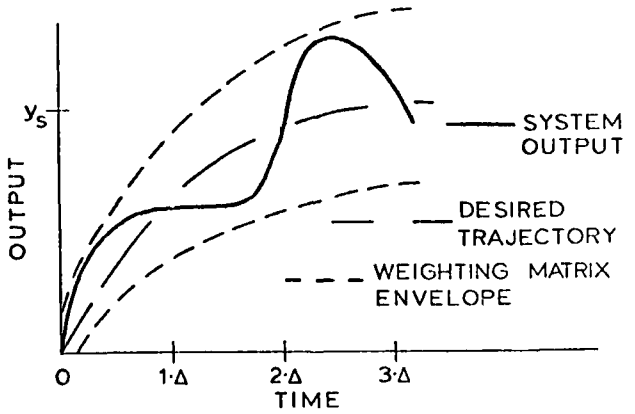
Fig 7. Controls Applied for Tau=0.3



a. Uniform Weighting About Desired Trajectory



b. Closer Fit at End of Interval



c. Closer Fit at Beginning of Interval

Fig 8. Different Possible Output Constraints Defined by the Weighting Matrix Q (Ref 1:79,81-82)

formulated to find the optimal control input, the relative sizes of the elements of the Q matrix define the shape of a tube (MIMO) or envelope (SISO) along the desired trajectory. In the least squares problem formulation, the output of the system is optimized to stay within the envelope (tube), as indicated in Figure 8.

The weighting matrix used to generate the envelope illustrated by Figure 8a is the identity matrix:

$$Q1 = \text{Diagonal}[1,1,\ldots,1]_{(NSM*L)X(NSM*L)} \tag{36}$$

Several alternative matrices are used to achieve the general effect illustrated in Figure 8b.

$$Q2 = \text{Diagonal}[1,1,..,1\,|\,2,2,..,2\,|\,4,4,..,4\ \ldots]_{(NSM*L)X(NSM*L)} \tag{37}$$

where each sub-block is NSM elements long

$$Q3 = \text{Diagonal}[1,2,4,8,16,\ldots]_{(NSM*L)X(NSM*L)} \tag{38}$$

$$Q4 = \text{Diagonal}[1,1,..,1\,|\,10^6,10^6,..,10^6]_{(NSM*L)X(NSM*L)} \tag{39}$$

The progressive application of Q2 , Q3 and Q4 results in the tube being squeezed tighter along the latter points of the interval (i.e., the oscillation of the system output about the desired trajectory is being progressively constrained). For the regulator applied to the fourth order system discussed earlier, the use of Q4 results in the output closely approximating the desired path.

The weighting matrix denoted as Q4 is used as a "bench mark" for comparison of the system output and controls required for variations

of the Q matrix. Figures 9 through 17 show the system output and control inputs required for Q1 , Q2 and Q3 versus Q4's use as the weighting matrix. In the parameter identification boxes (upper right corner of each graph), Q1 is referred to as "Identity Weighting," Q2 is referred to as "Block Weighting," Q3 as "Geometric Weighting" and Q4 as "MOD 2 Weighting."

A desire to have the system output initially follow the desired trajectory very closely and allow deviations from the desired path to increase with time is indicated in Figure 8c. The weighting matrix formulated to do this is

$$Q5 = \text{Diagonal}[\ldots,32,16,8,4,2,1]_{(NSM*L)X(NSM*L)} \tag{40}$$

Figures 18 and 19 show the system output and controls with Eq (40) as the weighting matrix. Scaling of the plots masks the fact that the system goes unstable almost immediately.

The choice of the weighting matrix to be utilized is, of course, application dependent. Natural damping of the system, allowable peak overshoot and specified settling time are all factors in the selection of Q . All of these specifications are reflected in the envelope (tube) shape selected from Figure 8. In this regulator application, with a very lightly damped system model, a trade-off between "nice" system response and the control energy required is apparent.

For the regulator applied to a very lightly damped system, utilization of a weighting matrix similar to Q3 , Eq (38), or Q4 , Eq (39), gives the best results. Use of Eq (40) is clearly a "bad" approach.

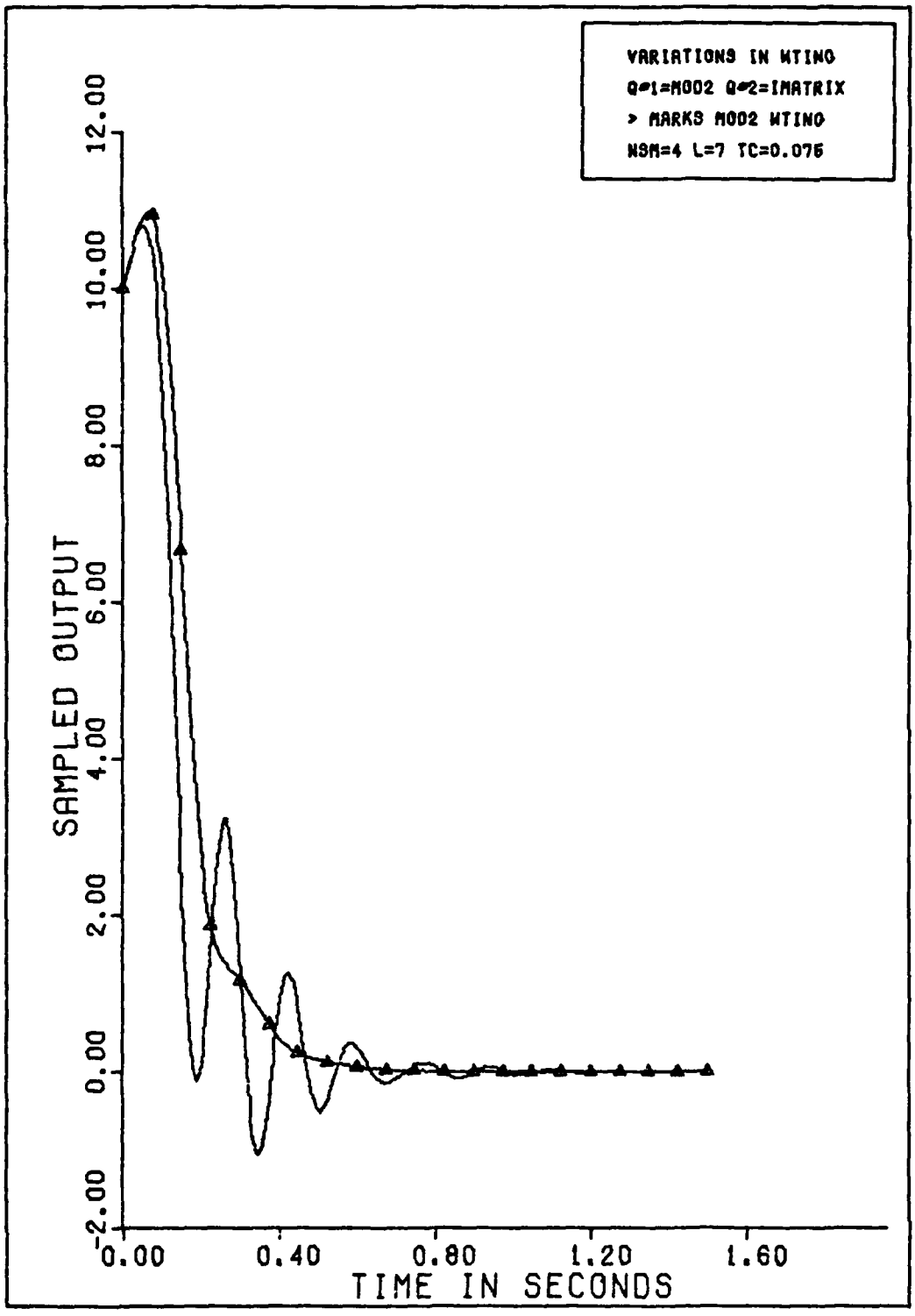


Fig 9. Sampled Output for Q4 (Δ) and Q1 Used as the Weighting Matrix

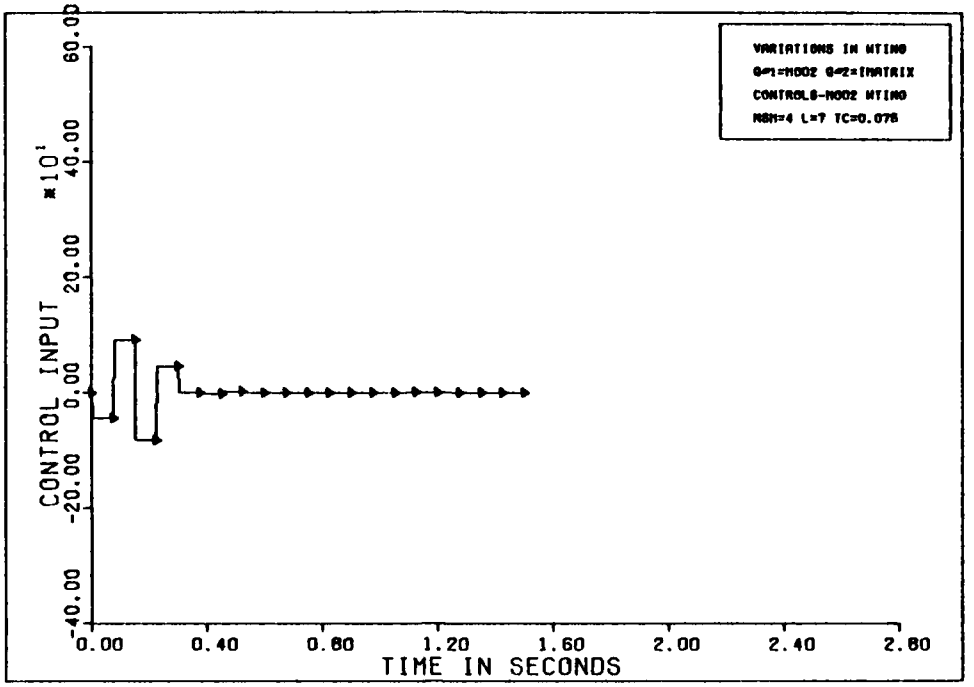


Fig 10. Controls Applied When Using Q4 as the Weighting Matrix

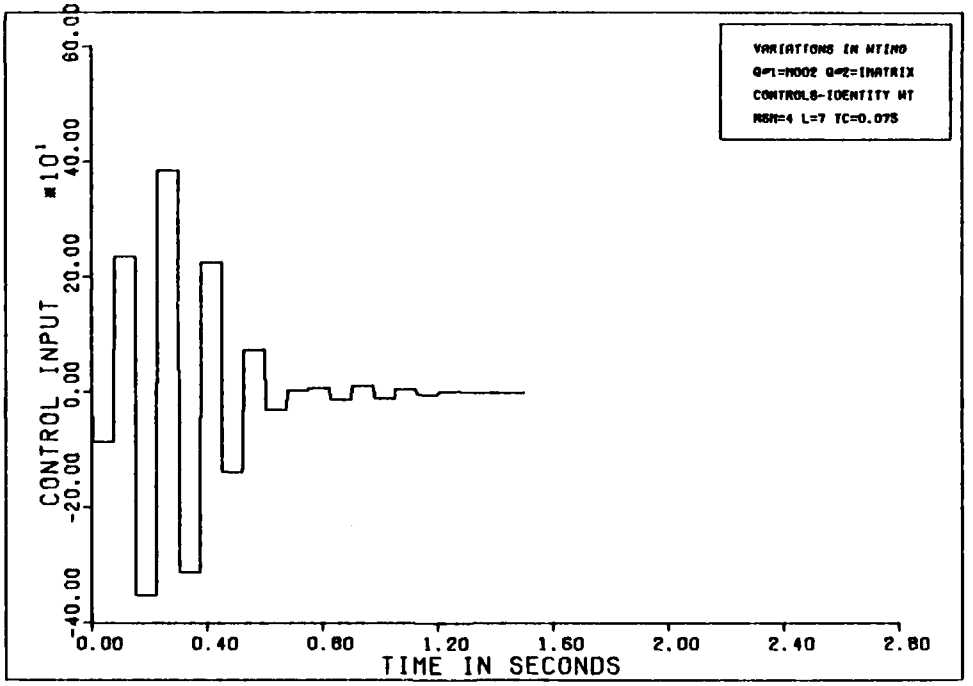


Fig 11. Controls Applied When Using Q1 as the Weighting Matrix

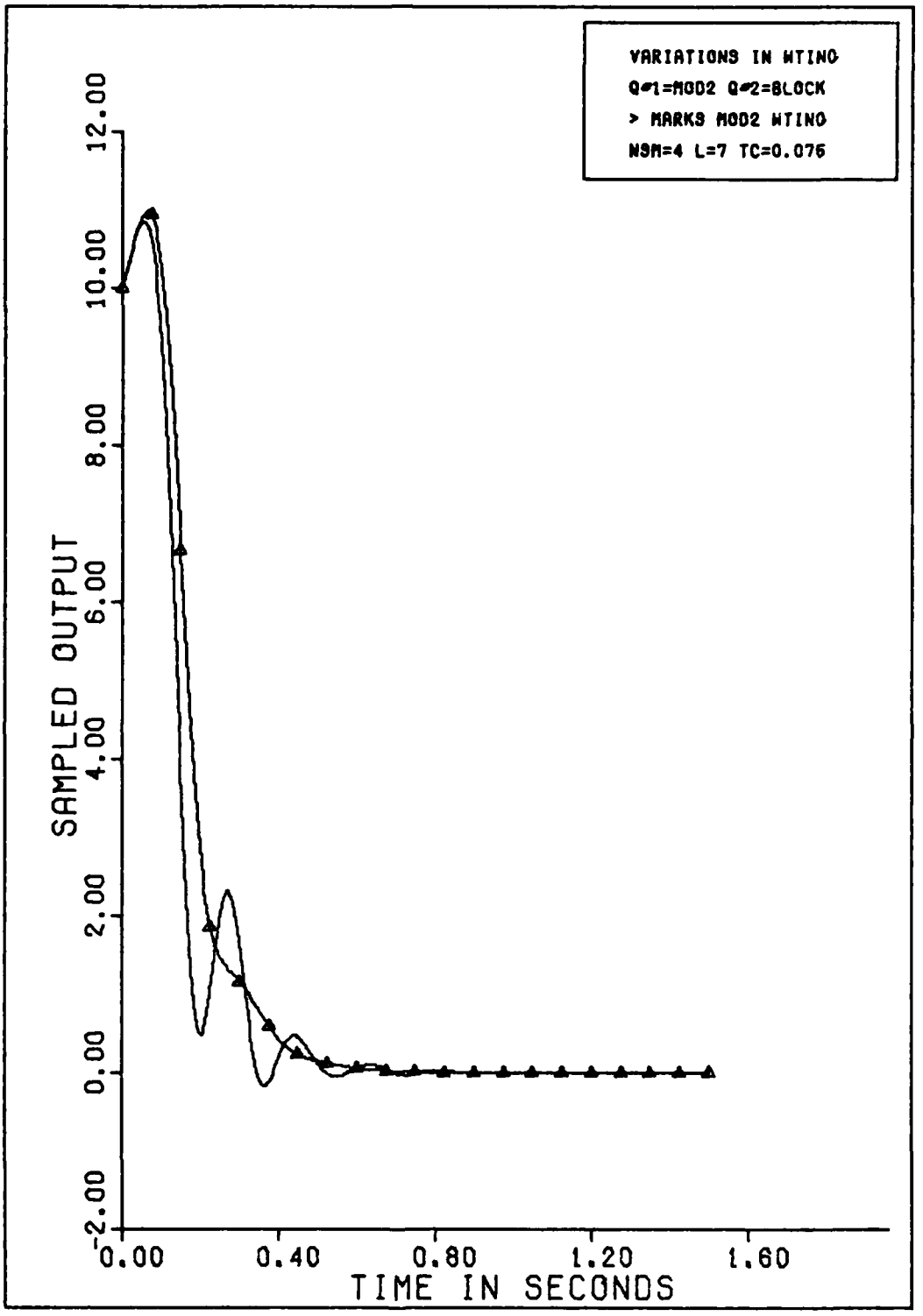


Fig 12. Sampled Output for Q4 (Δ) and Q2 Used as the Weighting Matrix

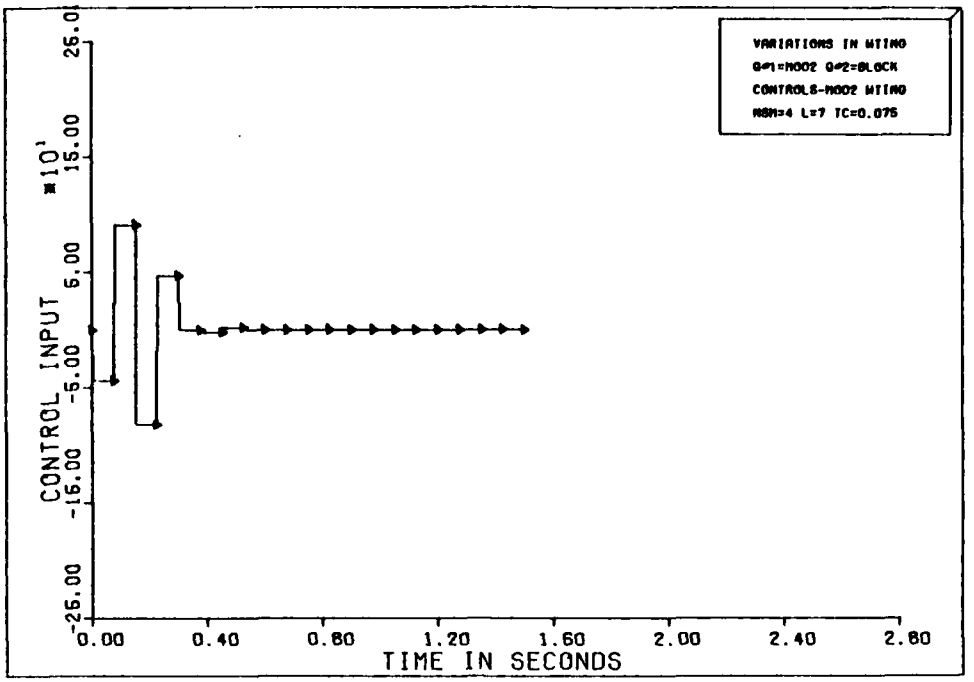


Fig 13. Controls Applied When Using Q4 as the Weighting Matrix

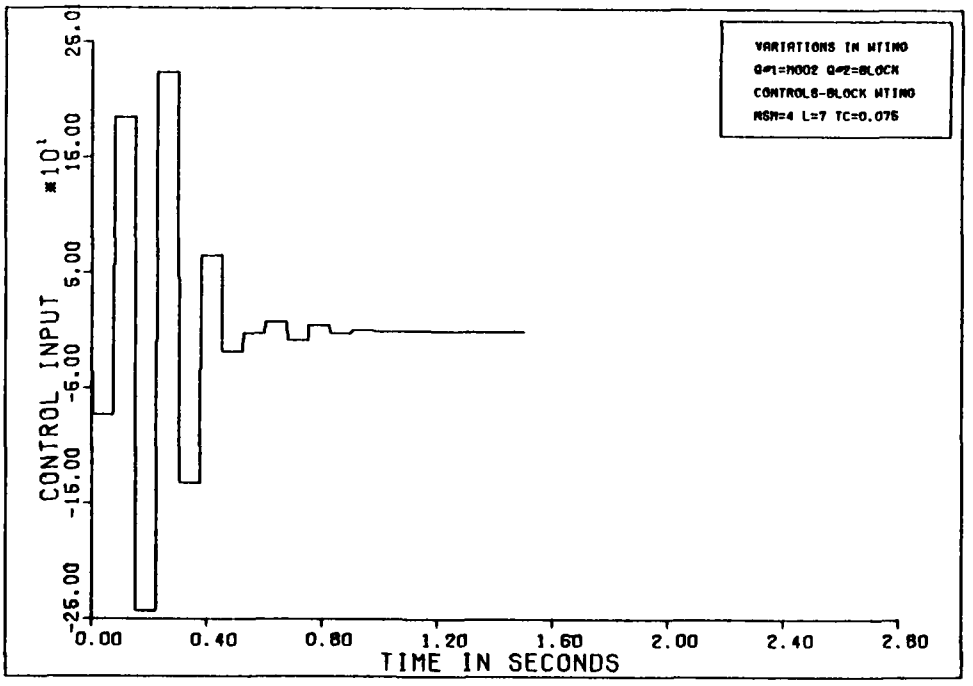


Fig 14. Controls Applied When Using Q2 as the Weighting Matrix

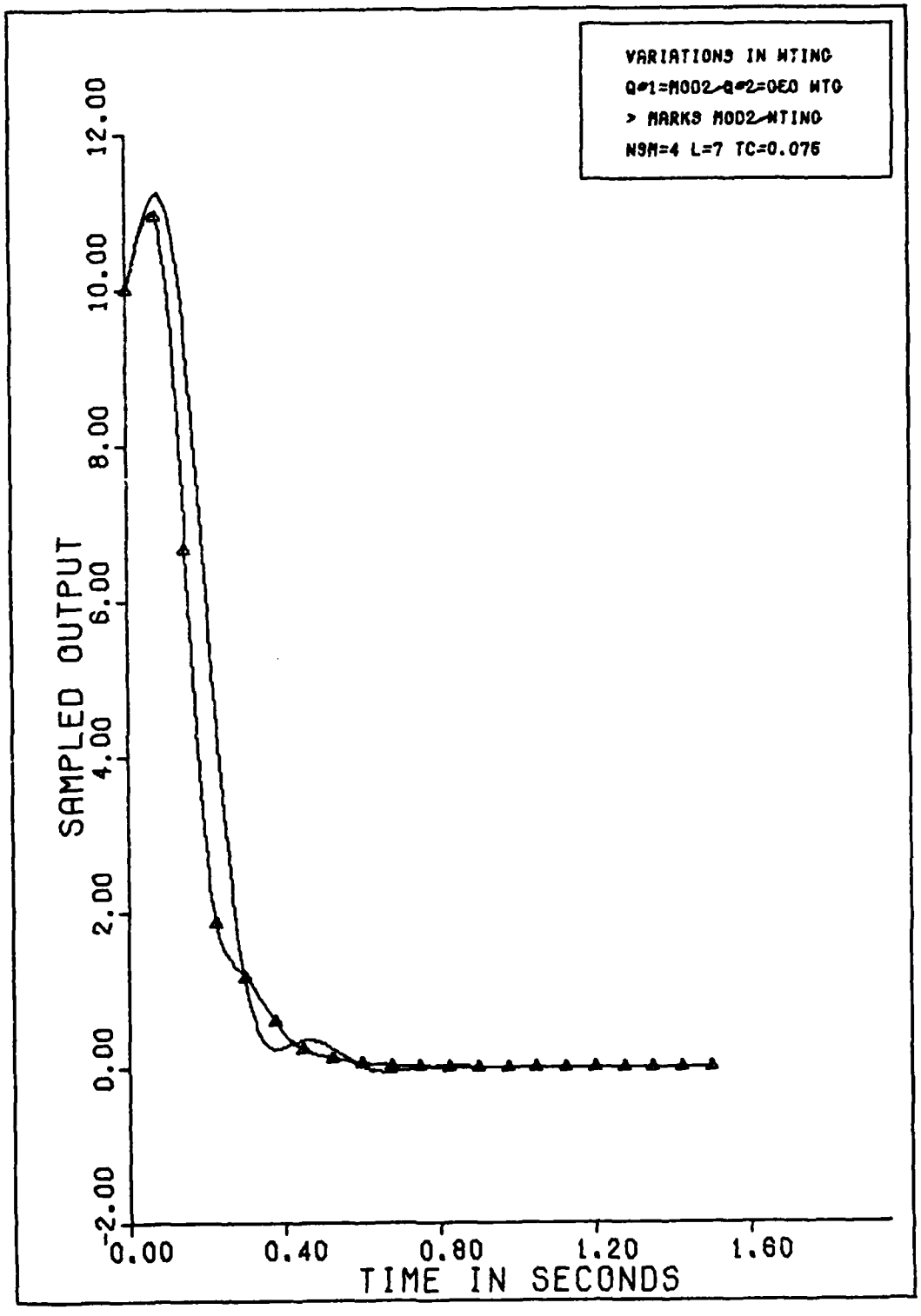


Fig 15. Sampled Output for Q4 (Δ) and Q3 Used as the Weighting Matrix

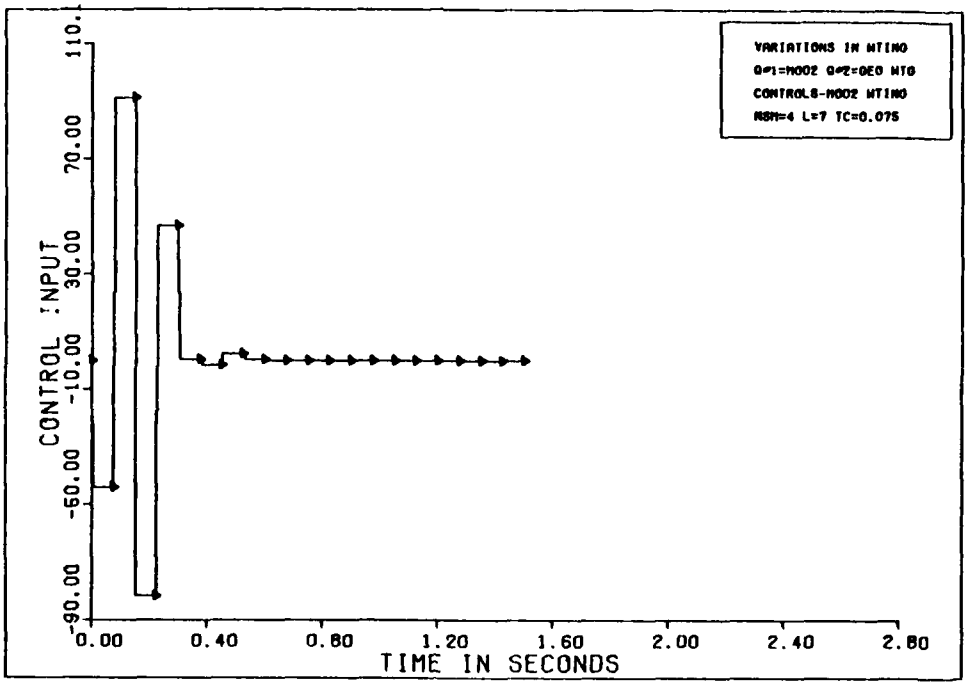


Fig 16. Controls Applied When Using Q4 as the Weighting Matrix

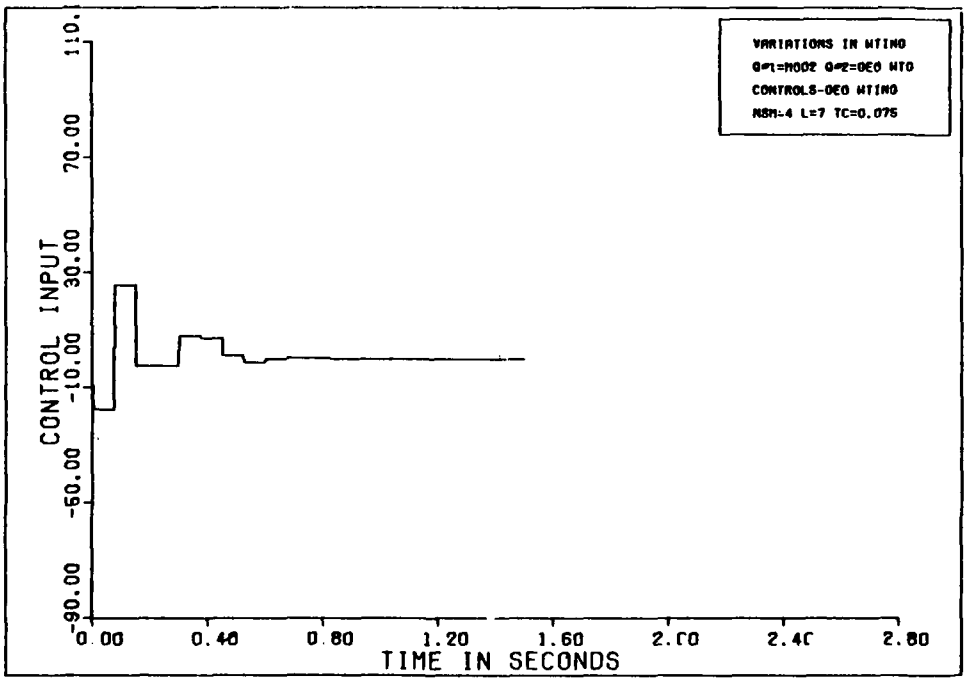


Fig 17. Controls Applied When Using Q3 as the Weighting Matrix

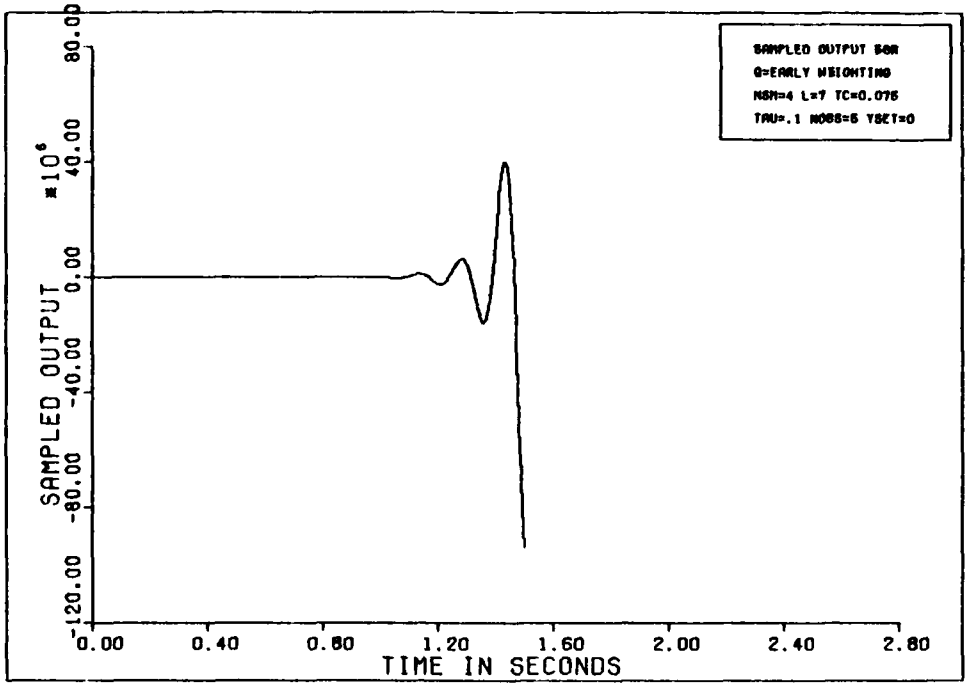


Fig 18. Sampled Output for Q5 Used as the Weighting Matrix

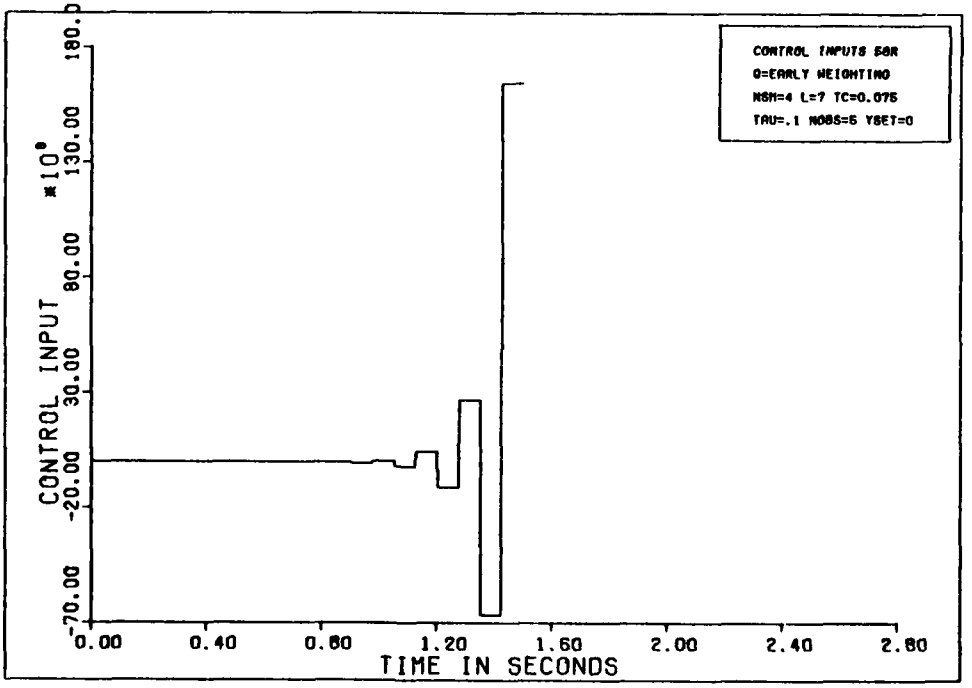


Fig 19. Controls Applied When Using Q5 as the Weighting Matrix

Variation of Number of
Smoothing Terms -- NSM

As the number of smoothing terms per control change is increased, the system response time increases, as seen in Figure 20. The initial increase from NSM=2 to NSM=4 smooths the response. However, a further increase to NSM=10 reintroduces oscillations in the output.

Figures 21 through 23 show the controls applied for each case. As NSM is increased, there is a substantial decrease (note factor of 10^{1} on controls for NSM=2) in the control energy required to bring the system to the zero set point. As the number of smoothing terms increases (i.e., the reference trajectory becomes better defined), less control energy is expended in controlling overshoots.

Variation of Number of Future
Controls Calculated -- L

The solution of the Normal Equation (28) is a vector of control inputs with length or dimension L . The quantity "L*TC" is a measure of how far in the future both prediction and calculations are carried out. The vector of controls could be sequentially applied; however, in this controller only the first element of the control vector is applied and the problem reformulated at each iteration. Figure 24 shows how variations in L affect the system output.

Increasing from L=3 to L=4 damps the large oscillations and increases the system response time. Also investigated (but not shown) is an increase to L=10 which further slows the system response and reintroduces some low amplitude oscillation.

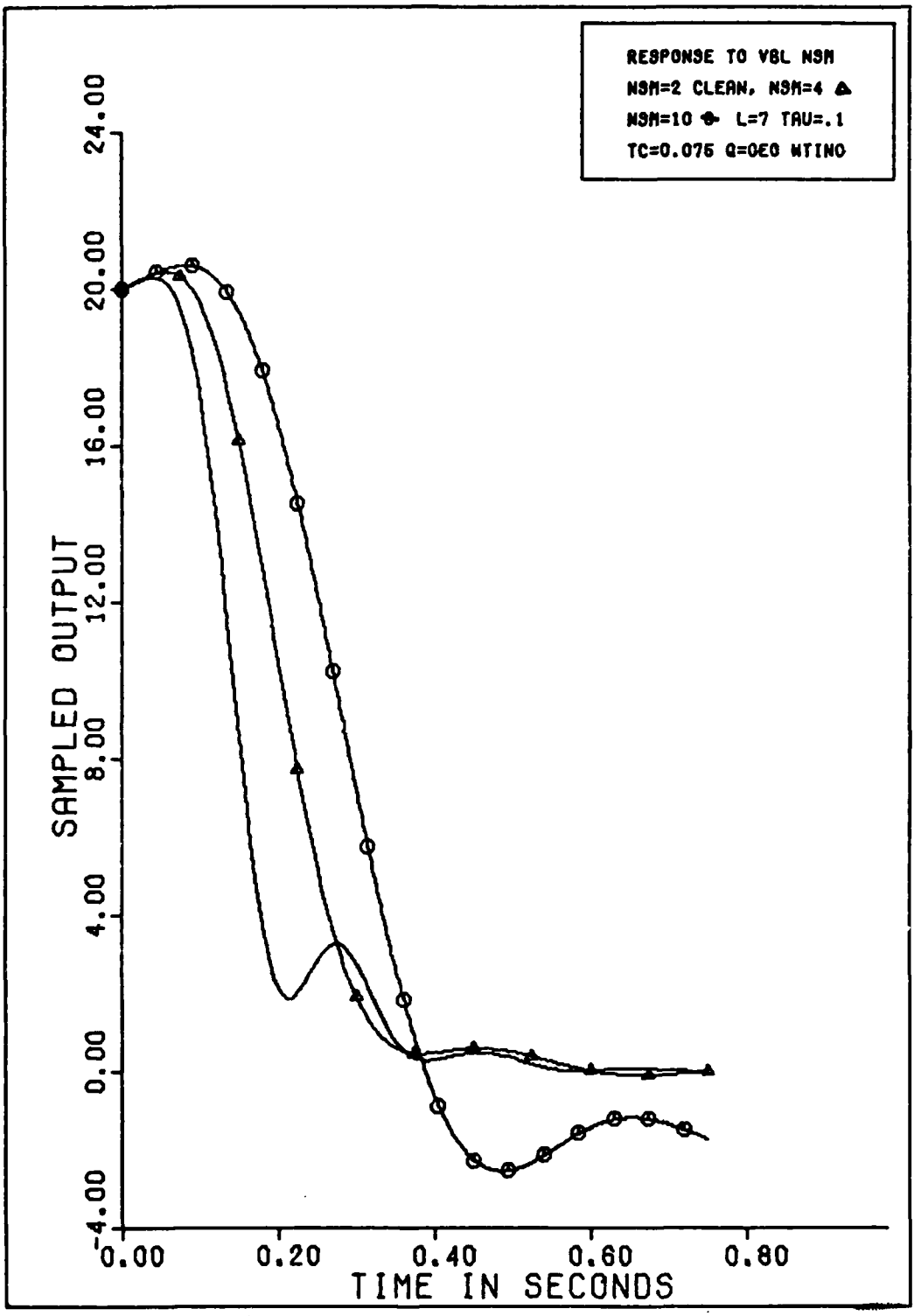


Fig 20. System Output for NSM=2 (Clean), NSM=4 (Δ) and NSM=10 (Ø)

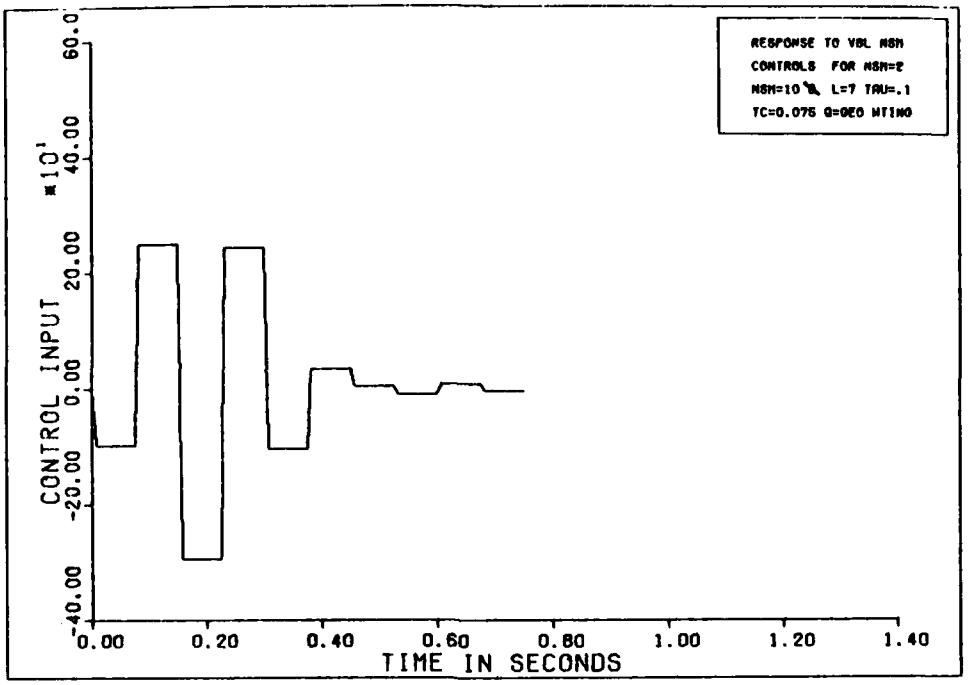


Fig 21. Controls Applied for NSM=2

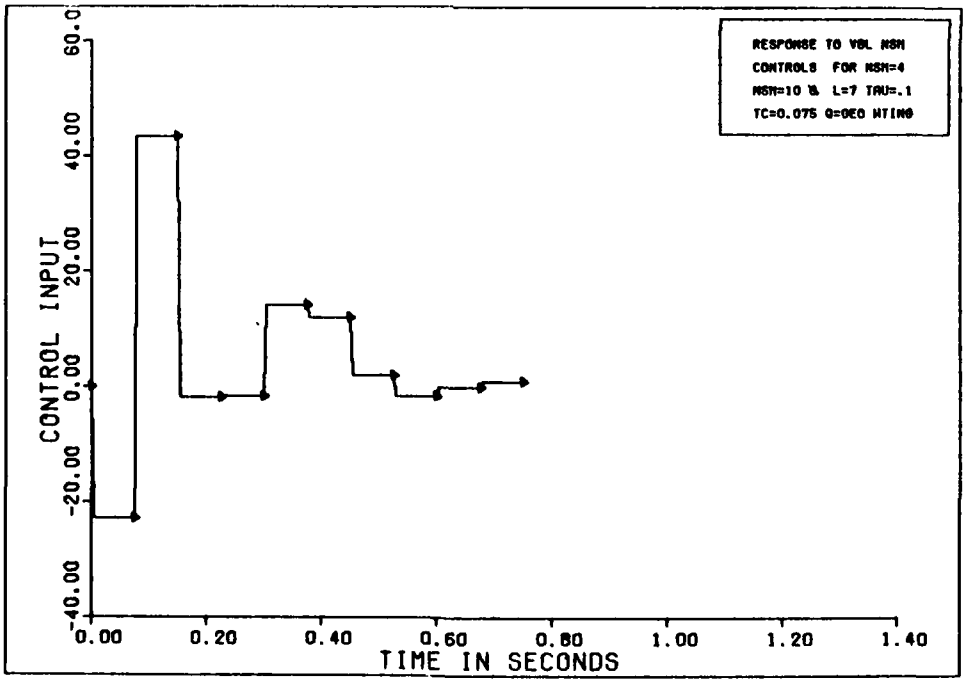


Fig 22. Controls Applied for NSM=4

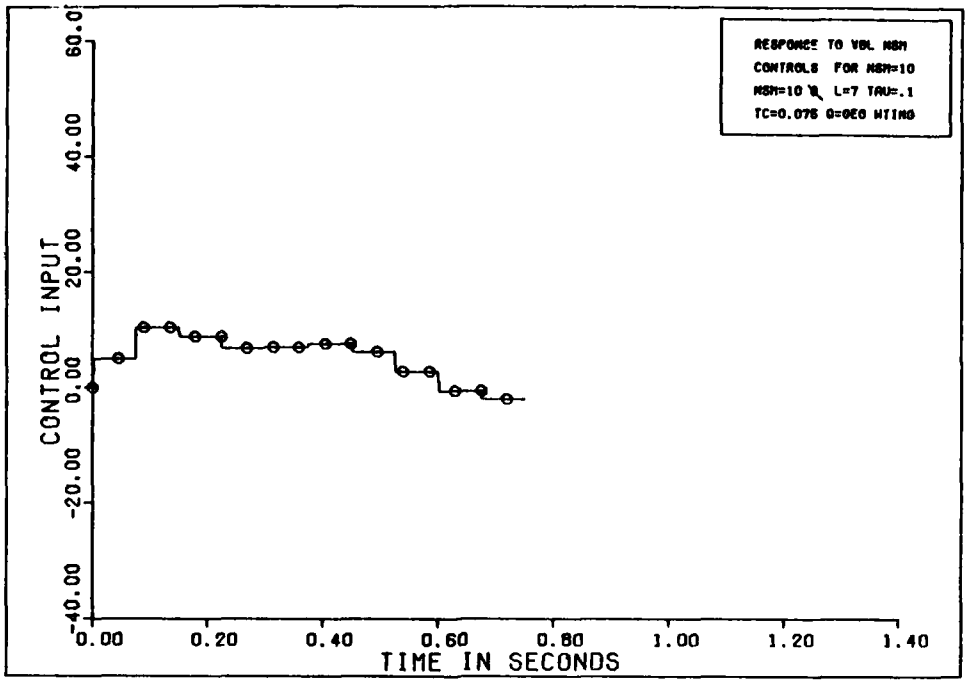


Fig 23. Controls Applied for NSM=10

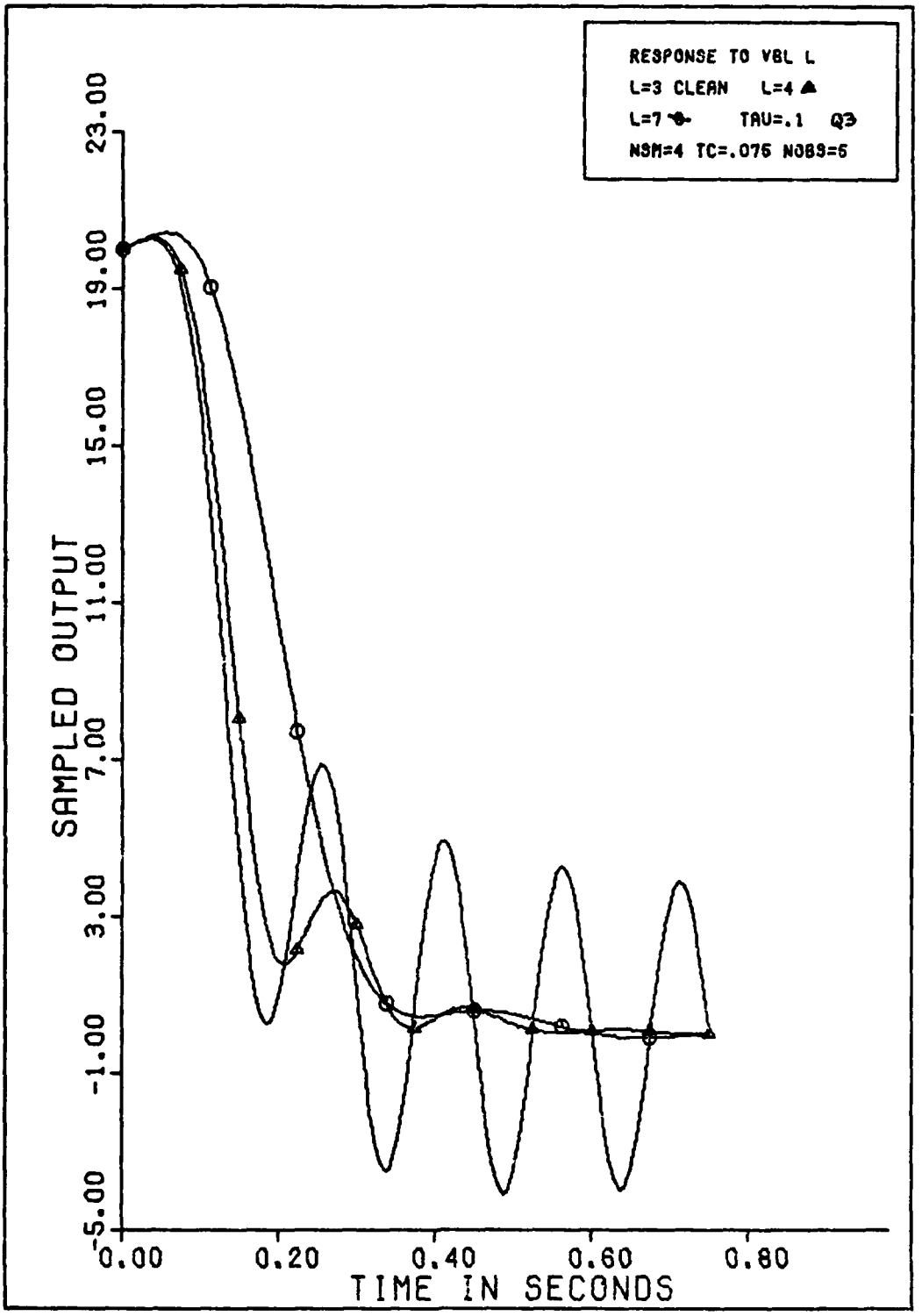


Fig 24. Sampled Output for L=3 (Clean), L=4 (Δ) and L=7 (Ø)

Figures 25 through 27 show that as L is increased, there is a decrease in the control energy required to bring the system to the zero set point.

The introduction of oscillations into the output for L>7 and large reduction in control energy required as L increases can be explained through an analysis of the weighting matrix (Eq (38)) utilized, and the fact that the time interval in question is increasing as L increases. Use of Q3 reflects a desire for a closer fit of the output to the desired trajectory toward the end of the time interval. This is indicated by Figure 8b. As the time interval of calculation is increased by increasing L , larger deviations are tolerated early in the time interval. As less weighting is given the early deviations, less control energy is spent trying to match the desired path during the period where it (the desired path) is rapidly changing.

A good initial guess for L is 2*N , where N is the dimension of the state vector. The final value of L chosen depends on the specifications for system response time and control input limitations.

Variations in the Time Span of Each Control -- TC

Other than a slight increase in system response time, Figure 28 indicates no real change in system output for variations in TC. There is, however, a dramatic change in control energy requirements as indicated by Figures 29 through 31.

A direct design procedure for selection of the "optimal" sample time to maximize closed loop robustness (robustness is discussed in Chapter IV) is developed in Ref 11. This technique was designed for

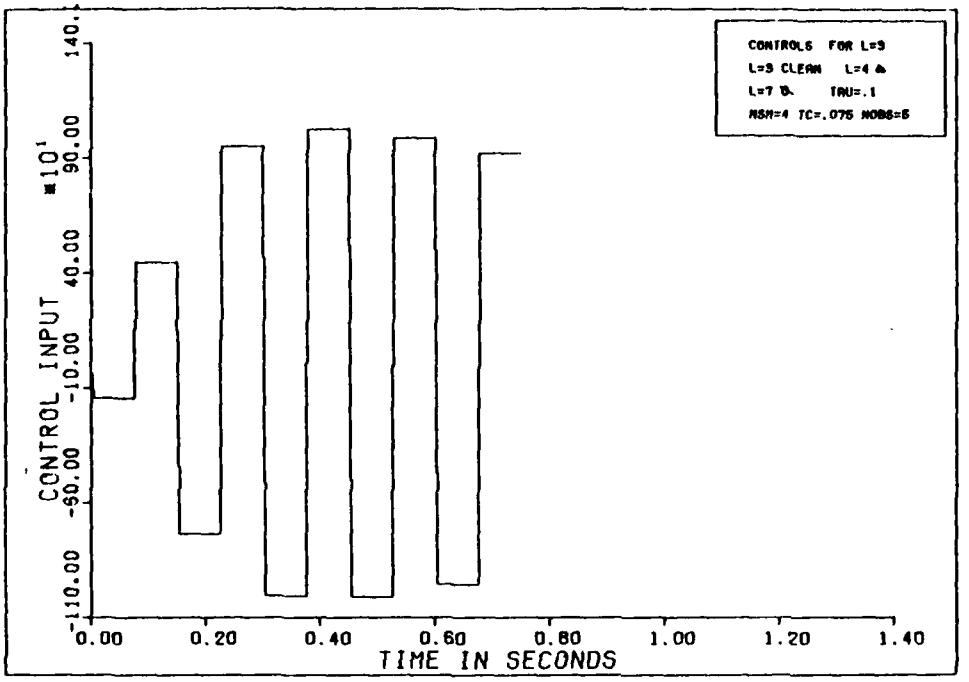


Fig 25. Controls Applied for L=3

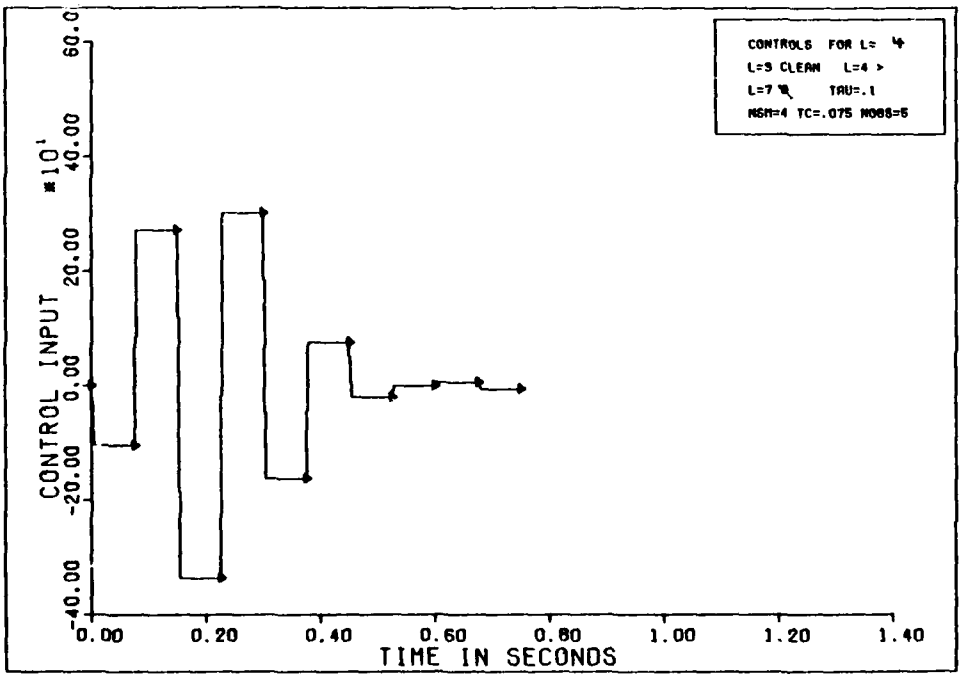


Fig 26. Controls Applied for L=4

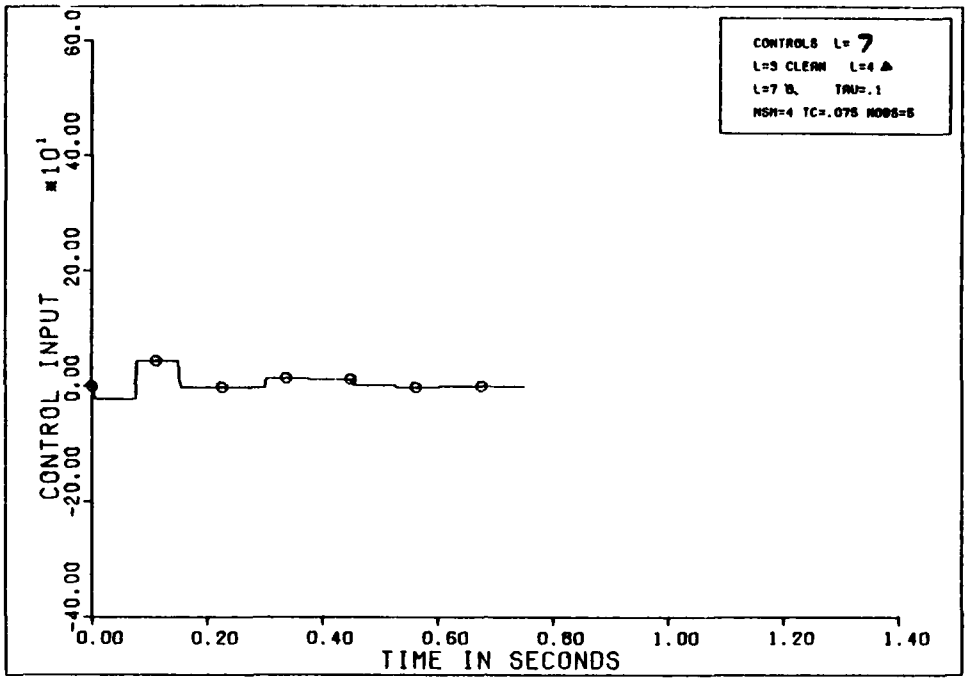


Fig 27. Controls Applied for L=7

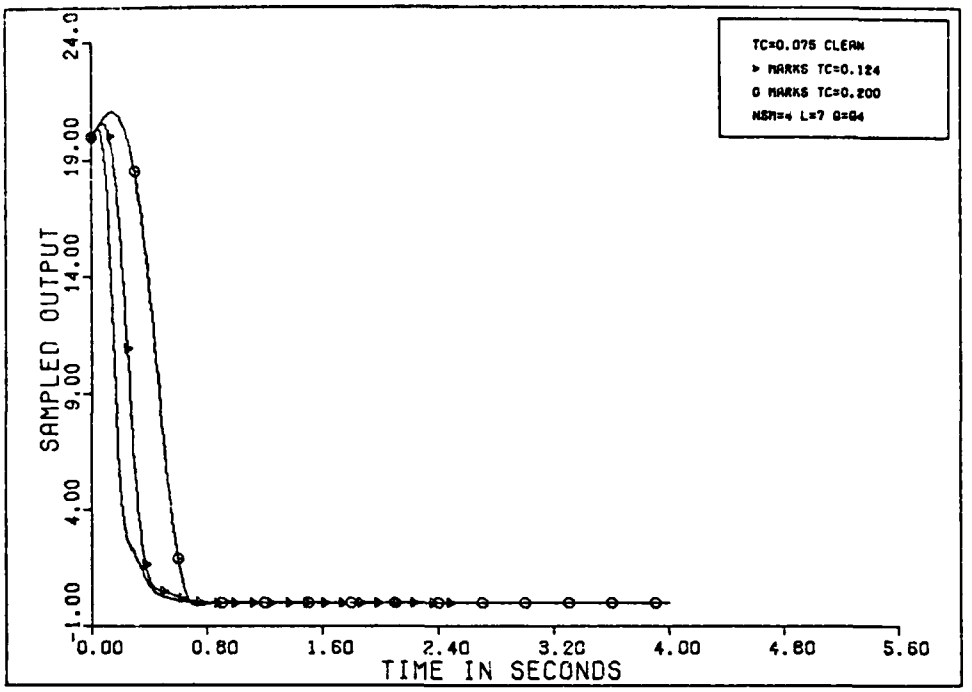


Fig 28. System Output for TC=0.075 (Clean), TC=0.124 (Δ) and TC=0.200 (Ø)

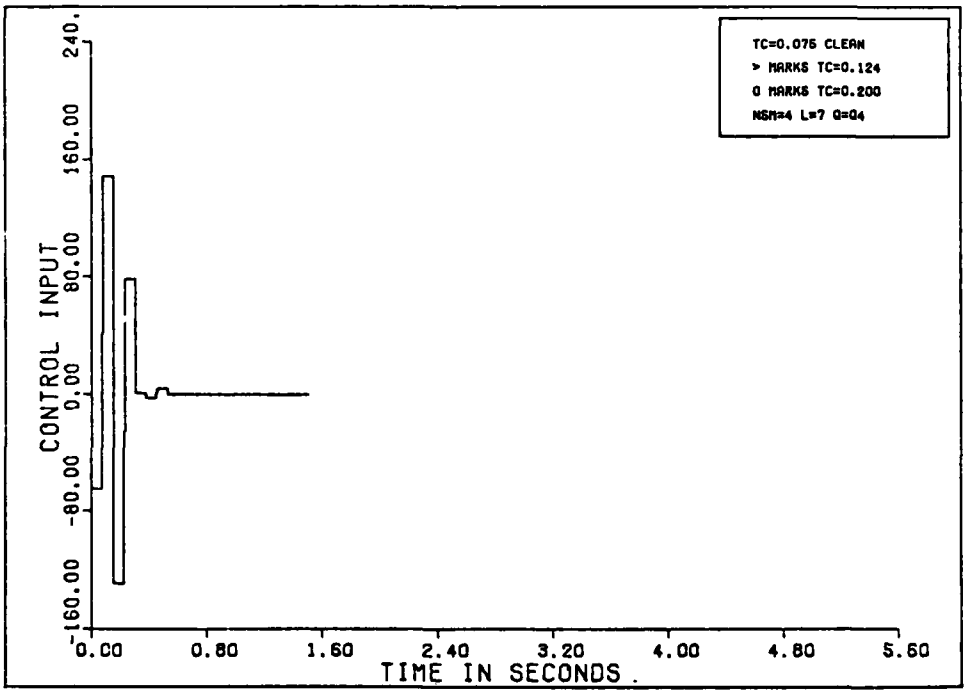


Fig 29. Controls Applied for TC=0.075

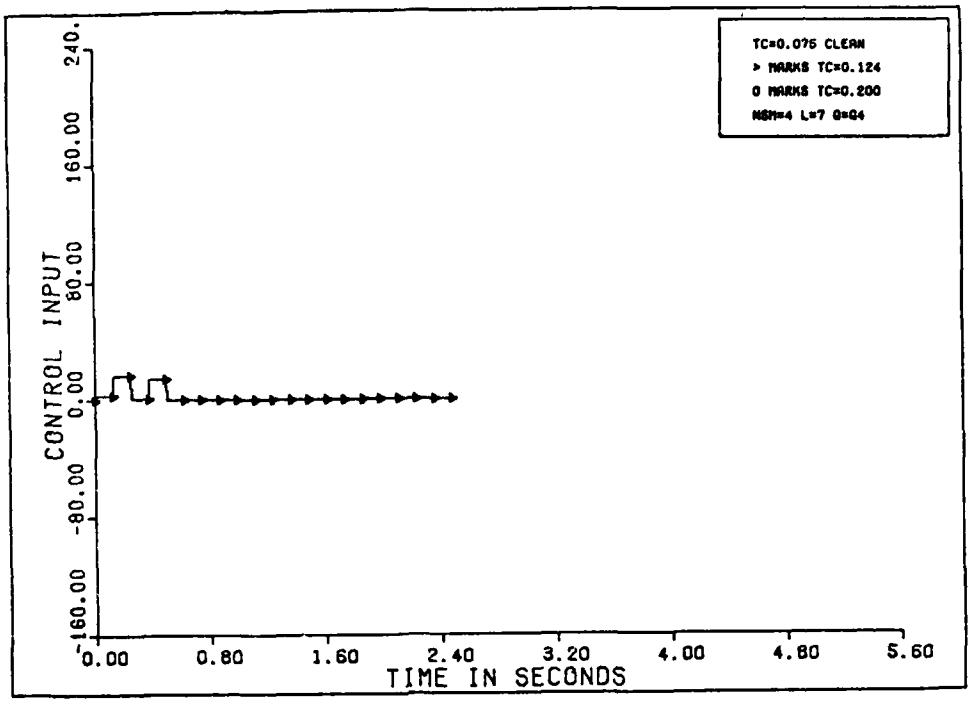


Fig 30. Controls Applied for TC=0.124

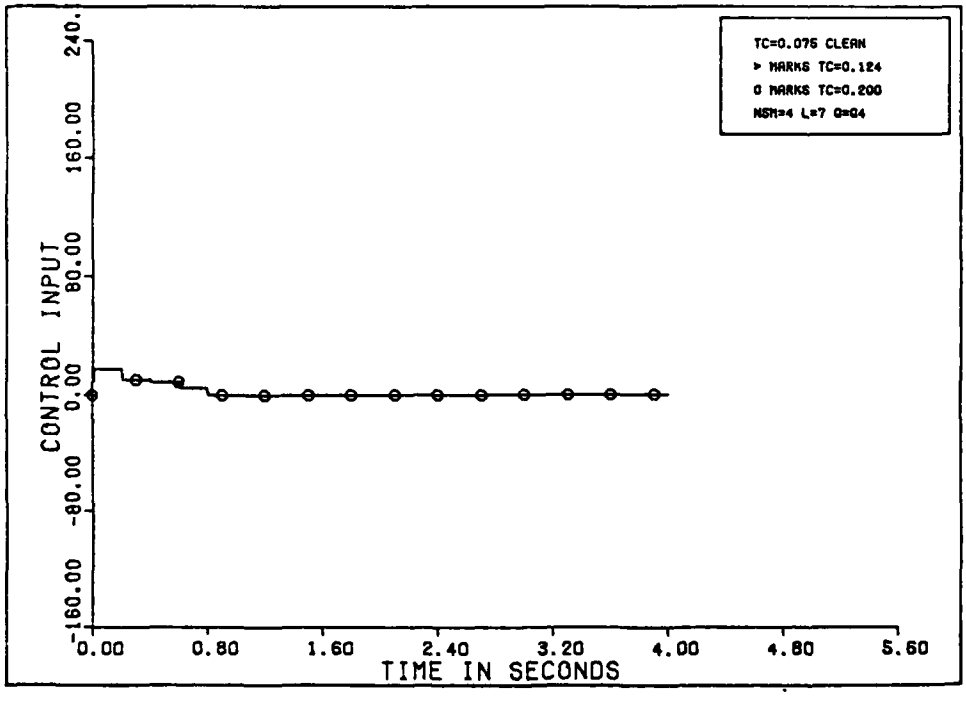


Fig 31. Controls Applied for TC=0.200

use with the Output Predictive Dead-Beat Controller (OPDEC) when investigating model mismatch. The basic procedure used is to investigate the reciprocal condition number, $1/\kappa$, of the Hankel matrix (H_n) as a function of sample time.

$$1/\kappa = \sigma_{min}/\sigma_{max} \tag{41}$$

where

$1/\kappa$ = Reciprocal condition number

σ_{min} = Minimum singular value (minimum magnitude eigenvalue of $H_n^T H_n$) of the Hankel matrix

σ_{max} = Maximum singular value of the Hankel matrix

Choice of the Hankel matrix for investigation stems from analyzing the equation used for calculation of the inputs for the OPDEC (Ref 11):

$$H_n \underline{u} = -\underline{y}_{zi} \tag{42}$$

where

H_n = Hankel matrix

$\underline{u}$ = Vector of control inputs

$\underline{y}_{zi}$ = Vector of zero-input responses

Using the same logic, the matrix in the Normal Equation to be investigated is $[\bar{H}'Q\bar{H}]$.

The sample time that maximizes $1/\kappa$ is hypothesized to be the "optimal" control change time (TC) for the minimization of perturbation effects (model mismatch, input and measurement noises) on the system. It is also a realistic initial choice for TC , without regard

to any perturbation of the original system. Figure 32 is a plot of $1/\kappa$ of $[\bar{H}'Q\bar{H}]$ for the system model. Relative peaks in $1/\kappa$ occur at approximately 0.088 and 0.124 with valleys at 0.075 and 0.200. Although Figure 28 shows no real change in system response for variance of TC , selection of TC=0.124 will yield a much better response when the "original" system model is in error (i.e., model mismatch). This will be shown in Chapter IV.

Nonminimum Phase Systems

A brief investigation of a nonminimum phase system (i.e., system zeroes in the right-half plan (RHP)) is discussed in this section. It is hypothesized that selection of Tau, Q and NSM is the same as for the basic system with no zeroes in the RHP; however, because of the inherent difficulty in controlling a nonminimum phase system, the controller should look further into the future (increase in L) for calculation of the controls. The reciprocal condition number of $[\bar{H}'Q\bar{H}]$ built from the nonminimum phase system's equation is used to find TC .

If a zero at S=+0.3 is added to the original system, the open loop transfer function (OLTF) becomes

$$\text{OLTF} = \frac{(s-0.3)}{(s+0.55\pm j6)(s+0.25\pm j15.4)} \tag{43}$$

Figure 33 is a plot of the reciprocal condition number of $[\bar{H}'Q\bar{H}]$ for the system model of Eq (43). Initially the dimension of $\bar{u}$ is fixed at L=7 , the "best" value for the original (minimum phase) system model. Figures 34 through 37 show the system output and controls for TC=0.125 , TC=0.180 (good sample times) and TC=0.205 (bad choice, according to

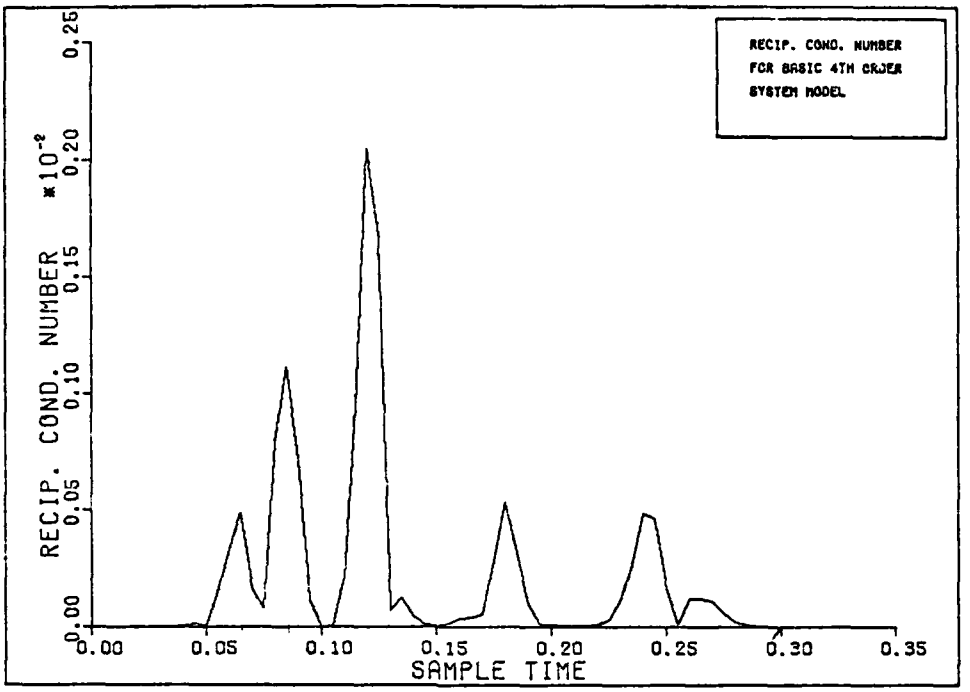


Fig 32. Reciprocal Condition Number of $[\bar{H}'Q\bar{H}]$ Using the Basic System Model

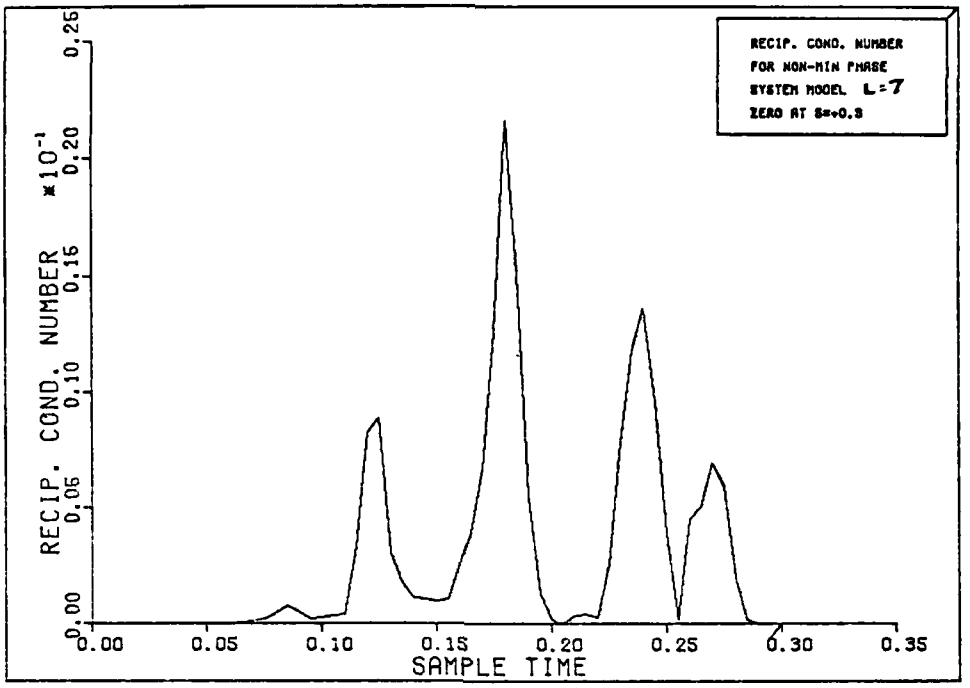


Fig 33. Reciprocal Condition Number of $[\bar{H}'Q\bar{H}]$ Using the Nonminimum Phase System Model

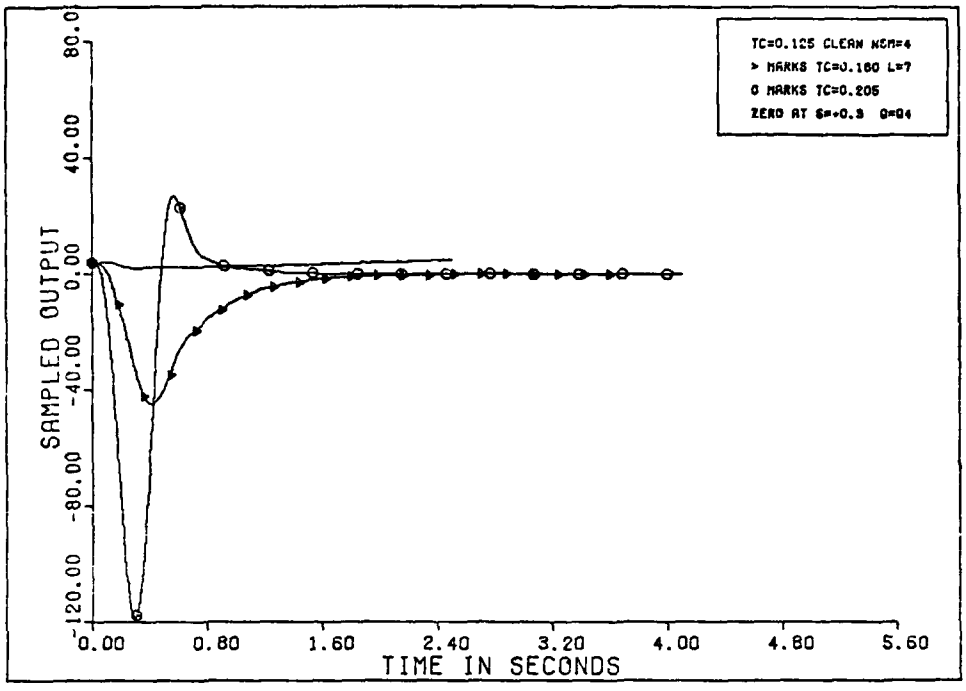


Fig 34. Sampled Output of Nonminimum Phase System for TC=0.125 (Clean), TC=0.180 (Δ) and TC=0.205 (Ø); L=7

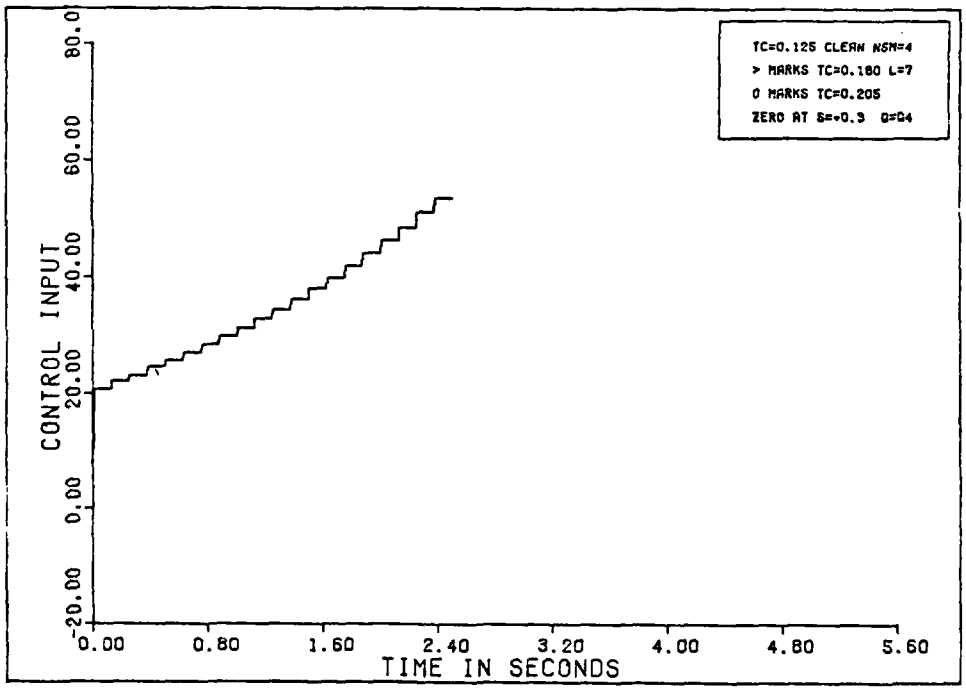


Fig 35. Controls for TC=0.125, L=7

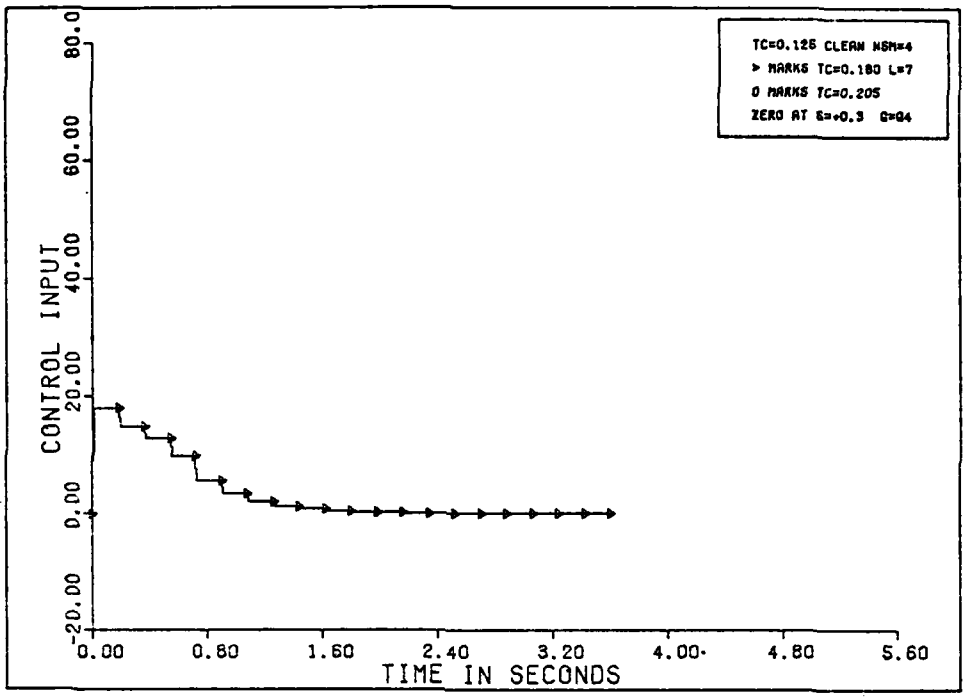


Fig 36. Controls for TC=0.180, L=7

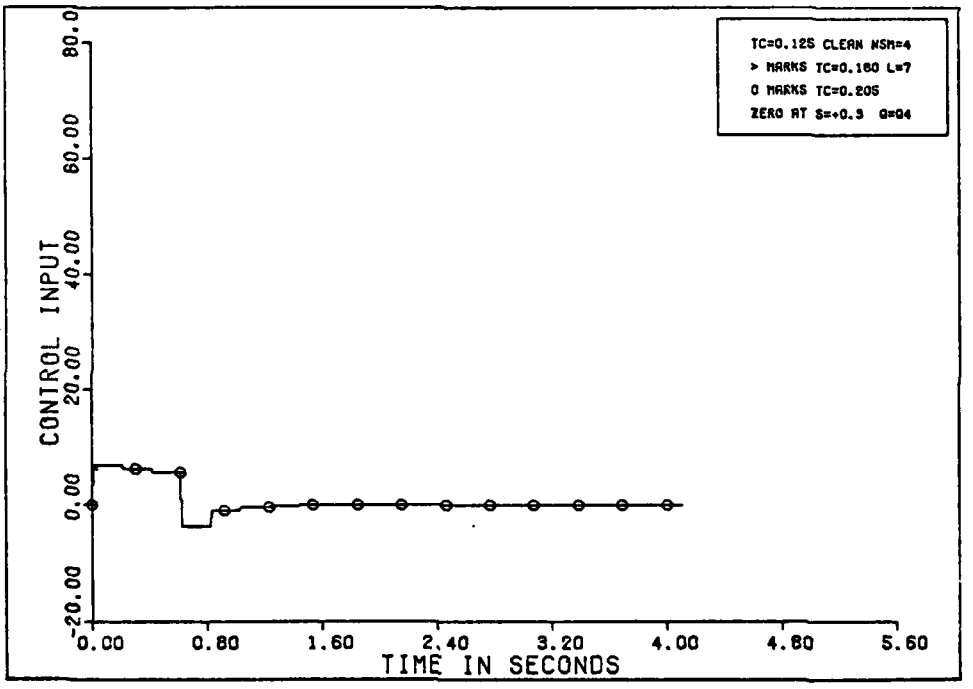


Fig 37. Controls for TC=0.205, L=7

Figure 33). An interesting observation is that for TC=0.125 , a supposedly "good" sample time for minimization of the effect of perturbation of the system model, the system goes unstable. This is easier to see in Figure 35, a plot of the controls for TC=0.125 . Although there is a large transient with TC=0.205 , the output is driven to (and kept at) the zero set point.

It is hypothesized that control of the nonminimum phase system is improved by allowing the controller to look further into the future. Figures 38 through 41 show the results of using the same sample times as above, but increasing the dimension of $\underline{\bar{u}}$ to L=10 . Instead of instability, using TC=0.125 results in the "best" response of the three sample times.

This limited analysis has not addressed all of the questions concerning control of the nonminimum phase system; however, the main factor in successful application of the hypothetical regulator to a nonminimum phase system is an increase in L over that used to control the basic all pole system.

Summary

An ironclad synthesis technique is not possible at this stage of development of the controller; however, the following guidelines are presented for selection of the internal parameters for control of the all pole system:

1. Specifications for settling time, response time and maximum allowable overshoot can be reflected in the tube shapes (Figure 8) defined by the elements of the weighting matrix. For the regulator

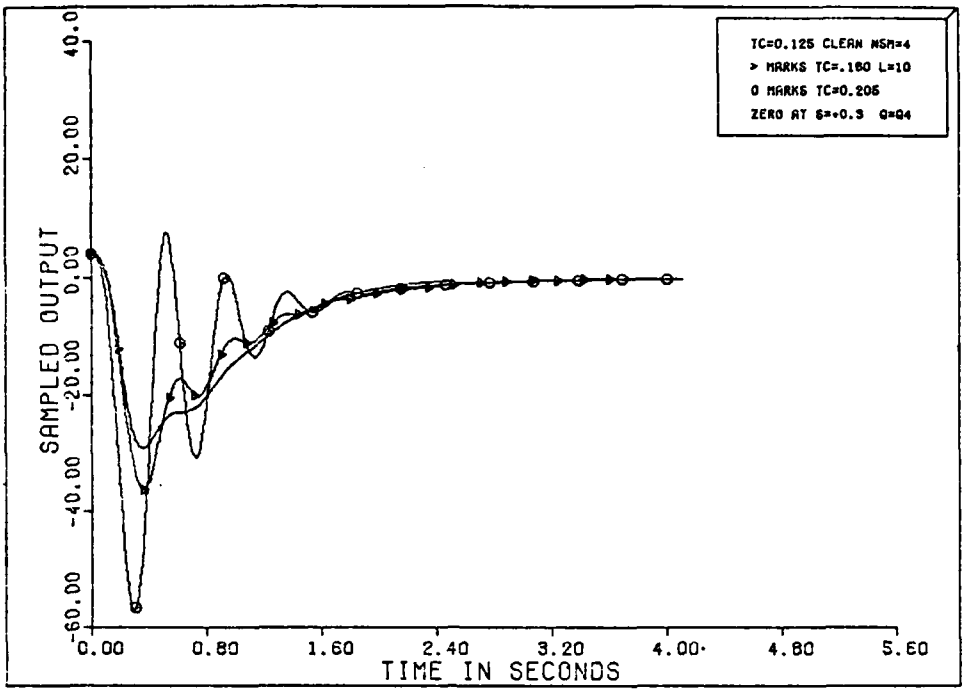


Fig 38. System Output of Nonminimum Phase System for TC=0.125 (Clean), TC=0.180 (Δ) and TC=0.205 (∅); L=10

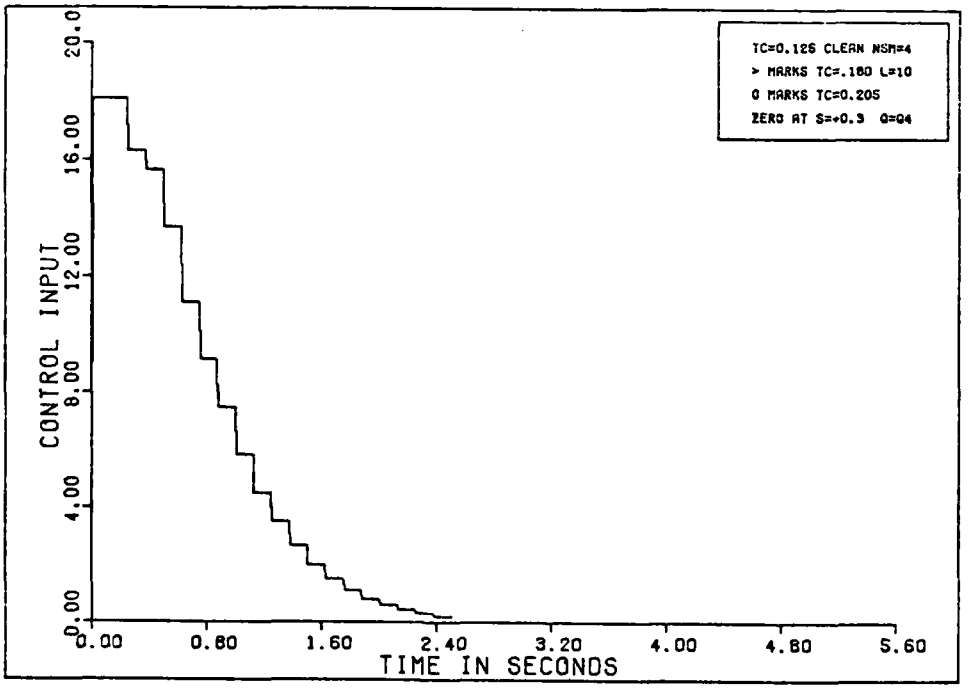


Fig 39. Controls for TC=0.125, L=10

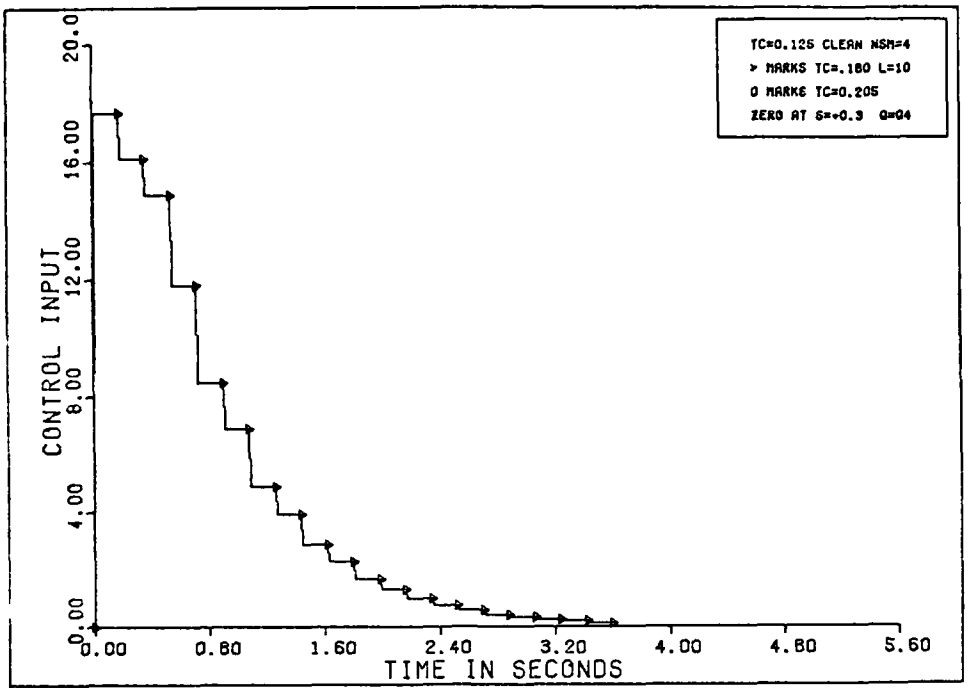


Fig 40. Controls for TC=0.180, L=10

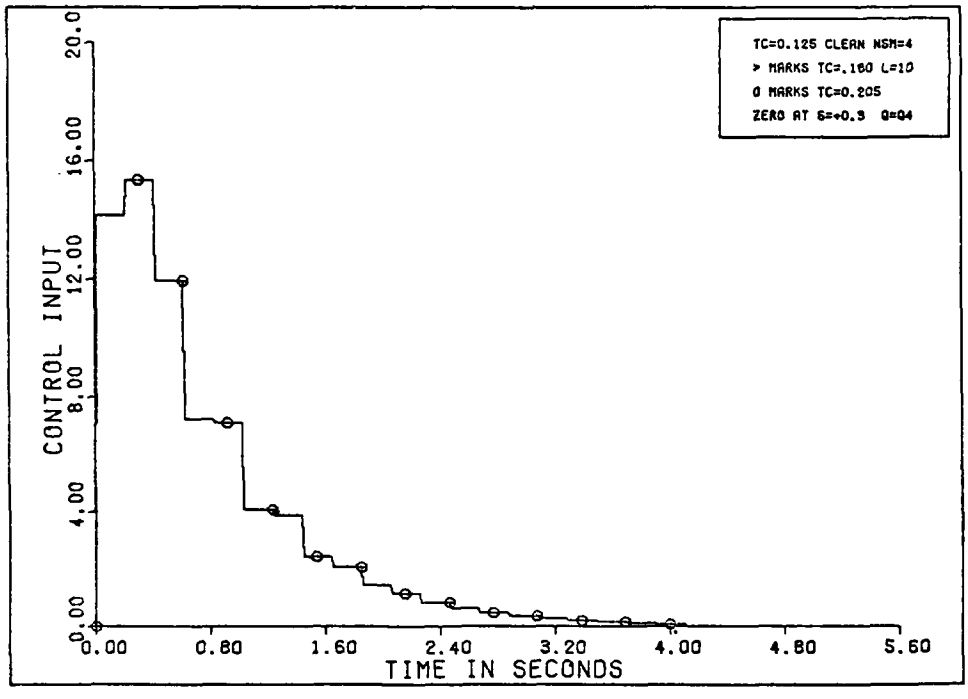


Fig 41. Controls for TC=0.205, L=10

applied to a lightly damped system, the weighting matrices identified in Eqs (38) and (39) gives a good system response. Eq (40), reflecting the tube shape indicated in Figure 8c is clearly a bad choice.

2. The sample time which maximizes the reciprocal condition number of $[\bar{H}'Q\bar{H}]$ is chosen as TC , the length of time a control is held constant. Although this selection process may not result in the best TC for an unperturbed system model, it is a good initial choice.

3. Tau is a reflection of how quickly the error between the desired and actual paths is brought to zero. Selection of Tau is dependent on the specified system response time. Its value will affect the control energy expended in taking the system output to the desired set point.

4. The number of smoothing terms (NSM) utilized depends on the shape of the desired path. A sufficient number of smoothing terms must be utilized to characterize the reference trajectory. For the first order exponential trajectory utilized in the regulator application, few smoothing terms are needed. As NSM is increased, resulting in a better defined reference path for solution of the least squares problem, the controls needed to drive the system output to the set point decreases. For the regulator example, $2 \leq NSM \leq 7$ works well.

5. An initial value for the number of future controls calculated (L) per iteration is 2n ; where 'n' is the order of the system model or dimension of the state vector. The final value of L chosen has a dramatic effect on control energy expenditures as well as system response time. L=7 is the choice for the regulator example. Variations from this introduces oscillations in the output.

For the nonminimum phase system, the same values for Tau, Q and NSM are applicable. The sample time which maximizes the reciprocal condition number of $[\bar{H}'Q\bar{H}]$ for the system equations is a good choice for TC . L must be increased over that used for the all pole system to facilitate control. Control of the nonminimum phase system is not fully understood at this point.

After an initial set of internal variables is selected, a scheme of parameter variation is necessary to get the best set of parameters for the particular application.

There are several interesting relationships between this "smoothing approach" and the Output Predictive Dead-beat Controller (OPDEC) discussed in Refs 3 and 11. Selection of L=2n and Q=Q4 given by Eq (39) for implementation of the "smoothing approach" is actually an approximation of the scheme used in the OPDEC approach and gives much the same kind of performance as OPDEC itself.

IV Perturbations of the System Model

For many practical problems, the true system model is unknown or the system dynamics change with environmental conditions. An example of such a system is an aircraft for which the flight characteristics change as a function of altitude, airspeed, attitude or some other parameter. Either of the above situations can be considered a case of model mismatch; that is, the actual system equations differ from the system model used in designing the controller. In this thesis, model mismatch is simulated by perturbation of the eigenvalues of the plant or "A" matrix. The technique for selection of the best value of TC (see Chapter III) for closed loop "robustness" is tested in the model mismatch simulation.

The original example system model developed in Chapter III is used in this chapter. The controller is applied as a regulator with the control objectives as listed in Chapter II. The initial state and output relationship are as given by Eqs (31) and (32).

Robustness

A robust controller is one that continues to perform properly despite system perturbations (i.e., model mismatch and noise corrupted information fed back to the controller). For an analytical discussion of robustness, see Refs 7 and 11. For this analysis robustness is measured in terms of stability only.

Model Mismatch

For discussion of model mismatch, several system models are defined - the Truth model and Delta models. The Truth model is the original system represented by Eq (33). A Delta model is formed by changing the natural frequencies associated with each of the eigenvalues of the original system by 10%, 20% or 30%. Table I lists the OLTFs for each model.

TABLE I

System Models Utilized in Model Mismatch Analysis

System Model	OLTF
Truth	$\frac{1}{(s+0.55\pm j6)\ (s+0.25\pm j15.4)}$
Delta 1 (+10% eigenvalue change)	$\frac{1}{(s+0.605\pm j6.6)\ (s+0.275\pm j16.94)}$
Delta 2 (+20% eigenvalue change)	$\frac{1}{(s+0.66\pm j7.2)\ (s+0.3\pm j18.4)}$
Delta 3 (+30% eigenvalue change)	$\frac{1}{(s+0.715\pm j7.8)\ (s+0.325\pm j20.2)}$

The regulator is, of course, designed for use with the Truth model. Model mismatch occurs whenever the discrete matrix representation of one of the Delta models is used in calculation of the control and/or prediction of the zero-input response. One entry point into the control calculation is through the $\bar{H}$ matrix, given by Eq (22). Each

element of $\bar{H}$ is given by

$$h(i) = CF^{i-1}G \tag{10}$$

The zero-input response prediction is given by

$$y_{zi}(i) = CF^{i}\underline{x}(0) \tag{13}$$

Both $\bar{H}$ and y_{zi} are contained within the Normal Equation (28) used to find the input. In this analysis, however, they are considered separately as independent entry points for the discrete matrices describing the Delta models.

Model Mismatch Simulation Results

The basic regulator described in Chapter II is augmented with a series of logical switches (see Appendix A) allowing the Delta model matrices to be used for prediction of y_{zi} and/or calculation of $\bar{H}$. Tables II and III summarize the results of the simulation for selected values of TC. Figures showing the system output and controls applied are in Appendix C. Figures are included for those cases where the system is slowly going unstable.

Selection of the various values of TC for the model mismatch simulation was made with Figure 42 in mind. In Chapter III it is hypothesized that the discrete time which maximizes $1/\kappa$ of $[\bar{H}'Q\bar{H}]$ is the best TC for closed loop robustness of the controller. In relation to the value of TC, the following observations are made concerning Tables II and III:

TABLE II

Summary of Results for Perturbed or Delta Models Used to Calculate the Zero-Input Response

System Model	TC (sec)	Results
Delta 1	0.075	stable (Figure C-1)
(10% eigenvalue change)	0.088	stable (Figure C-4)
	0.124	stable (Figure C-7)
	0.200	unstable
Delta 2	0.075	stable (Figure C-10)
(20% eigenvalue change)	0.088	stable (Figure C-13)
	0.124	unstable (Figure C-16)
	0.200	unstable
Delta 3	0.075	stable (Figure C-19)
(30% eigenvalue change)	0.088	stable (Figure C-22)
	0.124	unstable
	0.200	unstable

TABLE III

Summary of Results for Perturbed or Delta Models Used to Find $\bar{H}$

System Model	TC (sec)	Results
Delta 1	0.075	stable (Figure C-25)
(10% eigenvalue change)	0.088	stable (Figure C-28)
	0.124	stable (Figure C-31)
	0.200	unstable
Delta 2	0.075	unstable (Figure C-34)
(20% eigenvalue change)	0.088	stable (Figure C-37)
	0.124	stable (Figure C-40)
	0.200	unstable
Delta 3	0.075	unstable (Figure C-43)
(30% eigenvalue change)	0.088	stable (Figure C-46)
	0.124	stable (Figure C-49)
	0.200	unstable

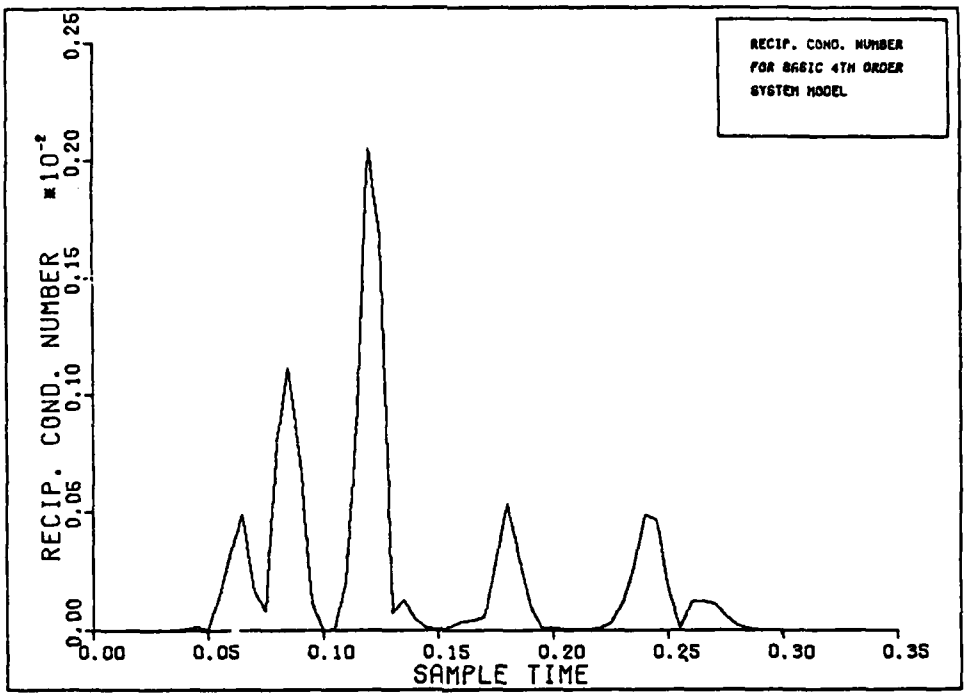


Fig 42. Reciprocal Condition Number of $[\bar{H}'Q\bar{H}]$ for the "Truth" or Basic System Model

1. From Table II, stability is maintained for all Delta models only with the smallest values of TC (0.075 and 0.088).

2. For TC=0.200 , the system is unstable for all Delta models to both $\bar{H}$ and y_{zi} .

3. From Table III, the only control change times for which stability is maintained for all Delta models correspond to the "peaks" in Figure 42.

As TC increases, the time span of the zero input response prediction also increases, thus amplifying the error involved when using a Delta model for prediction of y_{zi} . For any practical application of the controller, closed loop prediction would be accomplished through implementation of an observer or Kalman filter for use directly in the control law (17) or (28). This involves a transformation to an "output predictive state coordinate system," an idea developed by Dr. Reid in Ref 11. The new "state vector" is actually the predicted output (see Ref 11). When the zero input prediction responsibility is assumed by a Kalman filter, the results of Table II will be obviated. Selection of TC can be based on the information contained in Figure 42, substantiated by the results of Table III.

At TC=0.200, $1/\kappa$ is very small (see Figure 42), reflecting the fact that $[\bar{H}'Q\bar{H}]$ is ill-conditioned in terms of robustness. This is confirmed by Table III, in that for TC=0.200 the system is unstable for all Delta models sent to the calculation of $\bar{H}$.

TC=0.088 and TC=0.124 correspond to the two largest peaks in the reciprocal condition number, Figure 42. As seen in Table III, with these values of TC the system maintains stability with up to a

30% change in eigenvalues. Further, comparison of the system response for these two values of TC (Figures 43 and 44) indicates that the controller is "more robust" at the time which maximizes $1/\kappa$ (TC=0.124) .

Summary

The model mismatch simulation points out two important factors to be considered in a "real world" application. The first is that to avoid the results indicated in Table II, a Kalman filter or observer should be implemented to perform the future state estimation function. Second, in applications calling for a robust controller, the selection of control change time (TC) is a critical design step. The best choice for TC is the discrete time which maximizes the reciprocal condition number of $[\bar{H}'Q\bar{H}]$. The results demonstrated in this chapter concerning the selection of TC corresponds to the same result derived analytically in Ref 11 for the case of OPDEC.

This chapter concludes the discussion of the regulator application of the "smoothing approach" to Output Predictive Control. Chapter V discusses the technique implemented as a pitch controller on a modern aircraft in a terrain following problem.

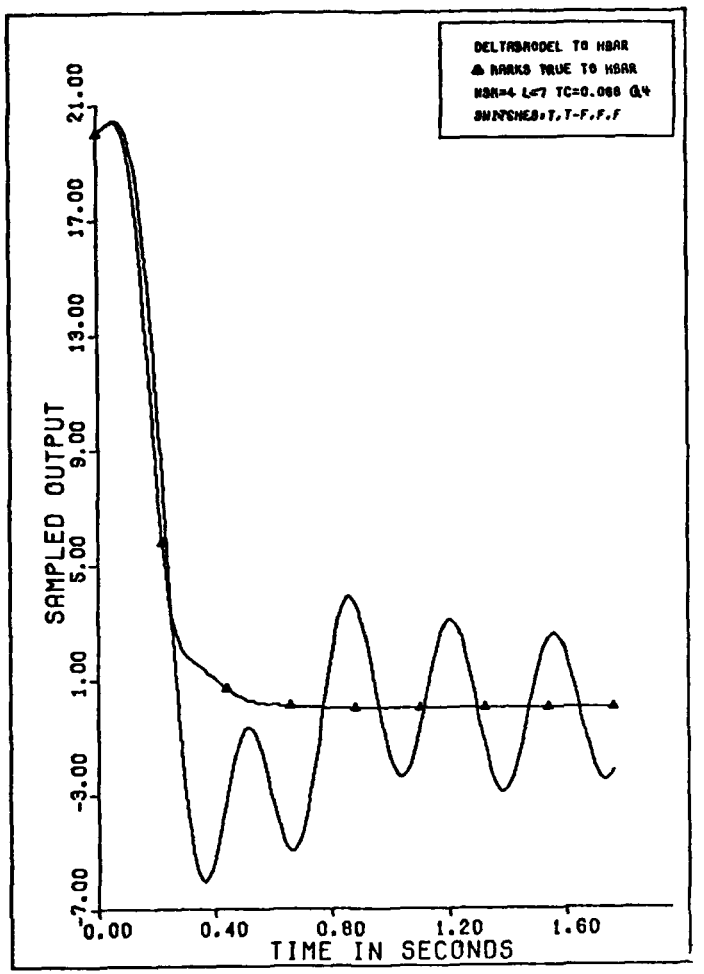


Fig 43. Sampled Output for Truth Model (Δ) and 30% Delta 3 Model to $\bar{H}$ Calculation, TC=0.088

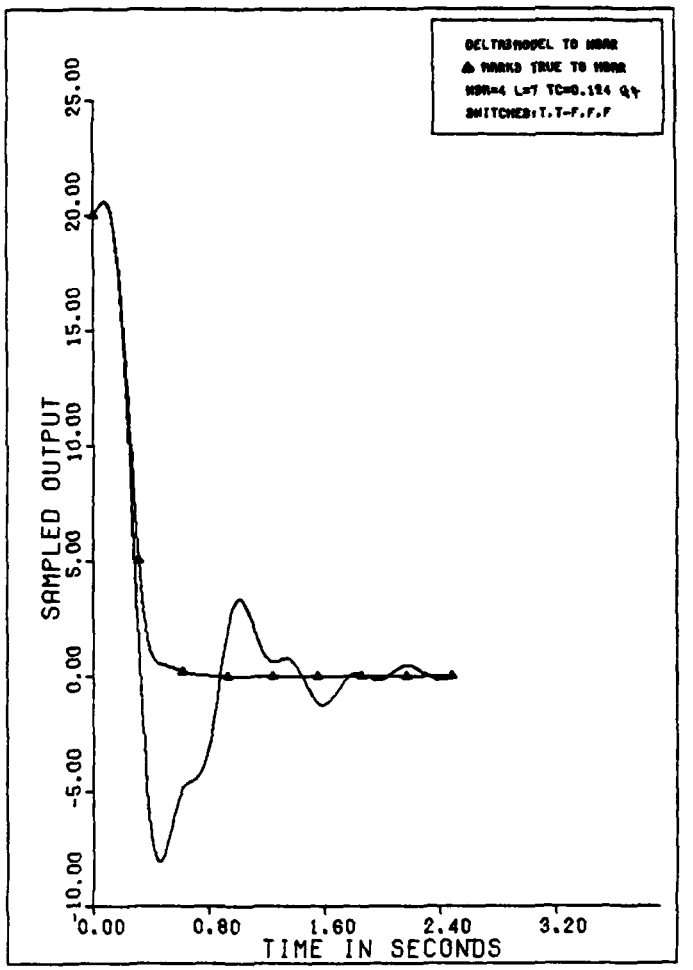


Fig 44. Sampled Output for Truth Model (Δ) and 30% Delta 3 Model to $\bar{H}$ Calculation, TC=0.124

V Controller Algorithm Implemented as a Pitch Axis Autopilot

In this chapter the algorithm developed in Chapter II is modified slightly for implementation as a pitch axis autopilot for a terrain following problem. The control objectives and aircraft model are discussed first, followed by an explanation of the changes necessary to the algorithm developed in previous chapters. Two test cases are presented and the results discussed. Model mismatch was not investigated. This development is not an attempt to present an ideal solution to the terrain following problem, but is included in the thesis as an initial investigation into a possible application area for the smoothing approach to Output Predictive Control.

Control Objectives

In Chapters III and IV the smoothing approach to Output Predictive Control is implemented as a regulator. In that case the set point is constant at zero for all time. In this chapter, the controller is used as a pitch axis autopilot and the objective is to follow a desired altitude profile; the set point becomes a time varying "set path."

System Model

The equations of motion for an aircraft are nonlinear with time varying coefficients. By assuming small perturbations about a specific operating point, however, a linearized set of equations is obtained. A linearized version of the longitudinal dynamics (called the short period approximation) for a modern fighter aircraft, linearized about a

straight and level operating point at 0.8 Mach, 10,000 feet altitude and v_o=constant=1077 fps , is given by

$$\begin{bmatrix} \dot{q} \\ \dot{\alpha} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} -3.854 & -12.240 & 0 \\ 1 & -2.834 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} q \\ \alpha \\ \theta \end{bmatrix} + \begin{bmatrix} 34.170 \\ .331 \\ 0 \end{bmatrix} \varepsilon \qquad (44)$$

where

q = Pitch rate (radians/sec)

α = Angle of attack (radians)

θ = Pitch angle (radians)

ε = Elevator deflection (radians)

The elevator deflection is assumed limited to $-0.262 \text{ rads} \leq \varepsilon \leq 0.426 \text{ rads}$.

For the short period approximation, ζ=0.69 and w_n=4.8 rad/sec.

Figure 45 shows the angles listed above and includes the flight path angle (γ), the angle between the horizon and flight path.

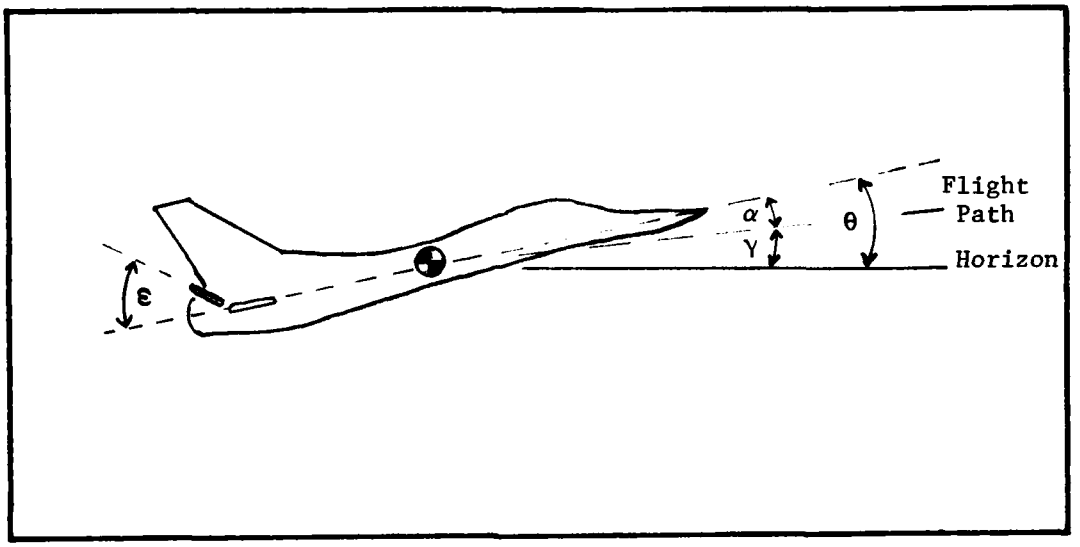


Fig 45. Perturbation Angles for Linearized Longitudinal Dynamics

With the forward velocity (v_o assumed constant in short period approximation) maintained constant through use of an automatic throttle system or pilot inputs, the additional state of altitude (h) is added through the relationship

$$\dot{h}(t) = v_o\gamma(t) = v_o[\theta(t)-\alpha(t)] \text{ fps} \tag{45}$$

where

$\dot{h}(t)$ = Vertical velocity (fps)

v_o = Constant = 1077 fps

The system model becomes

$$\begin{bmatrix} \dot{q} \\ \dot{\alpha} \\ \dot{\theta} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} -3.854 & -12.240 & 0 & 0 \\ 1 & -2.834 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & -1077 & 1077 & 0 \end{bmatrix} \begin{bmatrix} q \\ \alpha \\ \theta \\ h \end{bmatrix} + \begin{bmatrix} 34.170 \\ .331 \\ 0 \\ 0 \end{bmatrix} \varepsilon \tag{46}$$

The transfer function of interest is altitude change for a given elevator deflection

$$\frac{h(s)}{\varepsilon(s)} = \frac{-356.81(s-14.93)\ (s+18.78)}{s^2(s+3.34\pm j3.46)} \tag{47}$$

The controller, then, will use Eq (46) as the system model to calculate the elevator deflection necessary to follow a desired altitude profile.

Implementation

The algorithm used for the regulator is modified slightly for the terrain following problem. These changes include the generation of a set path, a modification of how the system error is reduced, the

addition of a control limiter and finally, selection of the internal parameters of the controller.

Set Path. Initially, the desired altitude profile or set path is chosen as a sinusoid with a variable frequency of oscillation:

$$y_{set}(i) = A\sin\omega t_i \tag{48}$$

From Figure 46, if A is fixed at 1000 feet, then the base or length of the "obstacle" can be calculated as

$$L = v_o t = v_o \frac{\pi}{\omega} \tag{49}$$

As the frequency of oscillation (ω) varies, the length (L) or base of the obstacle varies as given in Table IV. Figure 47 is a plot of the set path given by Eq (48) for A=1000 feet, ω=0.5 Hz and TC=0.5 sec .

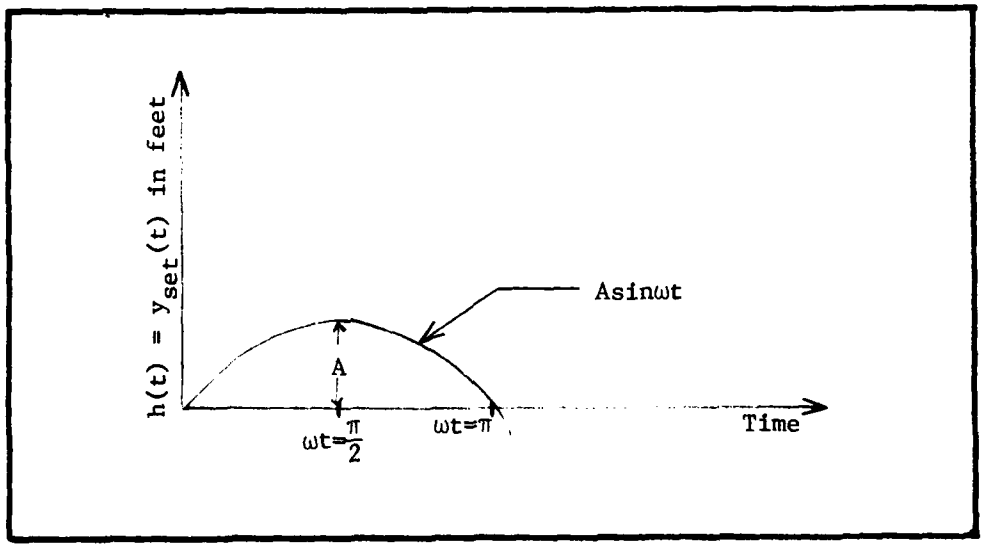


Fig 46. Asinωt as the Set Path

TABLE IV

Effective Size of Obstacle for Variation of the Frequency (ω) of the Sinusoidal Set Path

Frequency of Oscillation (ω) Hz	Base of Obstacle (ft)	Height (Ft)
0.50	6772	1000
0.60	5643	1000
0.75	4515	1000
0.90	3762	1000
1.00	3386	1000

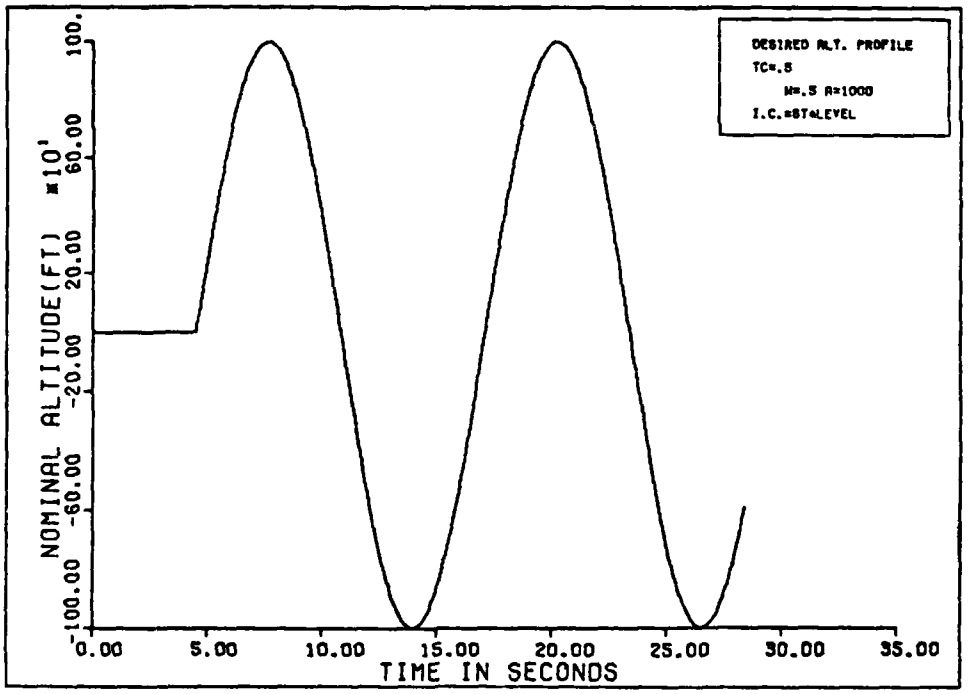


Fig 47. Set Path or Desired Altitude Profile for A=1000 ft, ω=0.5Hz, and TC=0.5 sec

Modification of System Error Reduction Relationship. In Chapter II, Eq (11) is given as the relationship for calculation of points along the desired trajectory from present position (y) to the set point ($y_{set}=0$ for regulator):

$$y_d(i) = y_{set}-\phi^i(y_{set}-y) \; ; \; i = 1,2,\ldots NSM*L \tag{11}$$

Eq (11) is based on the fact that, for the regulator application, the set point is constant for all time. A modification of this relationship is necessary because there is no longer a constant set point, but a time varying set path.

Assuming that an error exists between the actual system output and the set path, then $y_d(t)$ is a function of the present system output, present set point along the set path, future set points and how we wish to reduce the system error. If the error is to be exponentially reduced, then

$$\text{Desired Future Error} = \phi \text{ (Present Error)} \tag{50}$$

where ϕ is as given in Eq (11). Substituting into Eq (49)

$$y_d(t_{now}+iT_1)-y_{set}(t_{now}+iT_1) = \phi^i[y(t_{now})-y_{set}(t_{now})] \tag{51}$$

For the case of a constant set point where $y_{set}(t_{now}+iT) = y_{set}(t_{now})$, Eq (52) reduces to Eq (11). Eq (51) is used for calculation of discrete points along the desired trajectory.

Selection of Internal Parameters. As discussed in Chapter III, selection of the internal parameters is dependent on the physical

system. In this application of the algorithm, the parameters are chosen with physical restrictions in mind.

In an effort to use a physically realizable control change time, TC is chosen as $TC=0.5$.

Tau, the time constant of the decreasing exponential of Eqs (11) and (50), is a measure of how quickly the error between the desired and actual altitude profile is reduced. For this example, Tau is chosen as $Tau=0.3$. As Tau is decreased from this number, aircraft accelerations become excessive.

A sufficient number of smoothing terms (NSM) to give a well defined characterization of the desired or set path must be utilized. Values over the range $3 \leq NSM \leq 8$ were tried, with $NSM=4$ chosen. As the frequency of oscillation of the sine wave used as the set path increases, more smoothing terms per control are required to characterize this path.

Values of L , the number of future control changes calculated per iteration, over the range $4 \leq L \leq 10$ were tried. A value of $L=7$ is chosen to effect a good tradeoff between response time and the magnitude of the elevator deflections calculated.

The choice of a weighting matrix for the Normal Equation (28) is, of course, dependent on the values of L and TC . Each of the weighting matrices discussed in Chapter III and Eq (52) were tested as Q . Q6 is basically Q2 [Eq (37)] with the column order reversed and the nonzero elements moved to the principal diagonal:

$$Q6 = \text{Diagonal}[64,64,64,64\,|\,32,32,32,32\,|\,16,\ldots,1]_{28X28} \qquad (52)$$

where each block is NSM=4 units long. The resulting average and peak altitude errors for each weighting matrix are listed in Table V. Based on this data Q3 , Eq (38), is chosen as the weighting matrix to be utilized. Graphical results for the case of Q=Q4 and ω=0.5 are shown in Figures 48 through 49. It is interesting to note that although Q4 was the best choice for the regulator application, its use induces large altitude errors in this example.

Final Tests and Results

A sum of sinusoids is chosen as the set path or desired altitude profile for final testing of the controller:

$$y_{set}(i) = A\sin\omega_1(iT_1) + A\sin\omega_2(iT_1) + A\sin\omega_3(iT_1) \qquad (53)$$

The results of two test cases are presented. For both tests, A=1000 feet and the aircraft is started from an initial condition of straight and level flight. For the first case ω_1=0.5 , ω_2=0.6 and ω_3=0.2 ; for the second, ω_1 and ω_2 are unchanged and ω_3=0.8 . Table VI summarizes the results of the tests.

Figures 50 through 57 show the results of the test cases. Included for each test are plots of the desired and achieved altitude profiles, controls applied and pitch angle variations. Plots of the achieved altitude and control inputs show that the controller came to "steady state" tracking long before the inputs become steady state; this is considered a significant result.

Examination of the pitch angle history (Figures 53 and 57) for either case reveals angles well outside the range of "small perturbation"

TABLE V

Average and Peak Altitude Error for Variation of the Frequency of Oscillation (ω) of the Sinusoidal Set Path and Weighting Matrix Utilized

Weighting Matrix	ω(Hz)	Peak Error (ft)	Average Error (ft)
Q1	0.50	66.42	35.61
	0.60	79.49	40.65
Eq (36)	0.75	131.09	51.39
	0.90	263.81	74.42
	1.00	1051.37	160.80
Q2	0.50	68.17	35.38
	0.60	81.38	40.43
Eq (37)	0.75	164.04	51.84
	0.90	378.67	83.44
	1.00	1062.53	159.80
Q3	0.50	66.51	36.75
	0.60	76.96	42.35
Eq (38)	0.75	96.05	52.55
	0.90	113.70	61.98
	1.00	124.23	68.86
Q4	0.50	1007.79	100.04
	0.60	1007.79	104.12
Eq (39)	0.75	1007.79	114.60
	0.90	1007.79	126.70
	1.00	1007.79	136.03
Q5	0.50	333.72	74.52
	0.60	213.09	59.75
Eq (40)	0.75	1406.44	214.43
	0.90	1116.40	368.62
	1.00	6421.20	1766.51
Q6	0.50	67.52	36.01
	0.60	81.02	41.10
Eq (52)	0.75	101.15	51.09
	0.90	162.67	62.93
	1.00	703.38	161.68

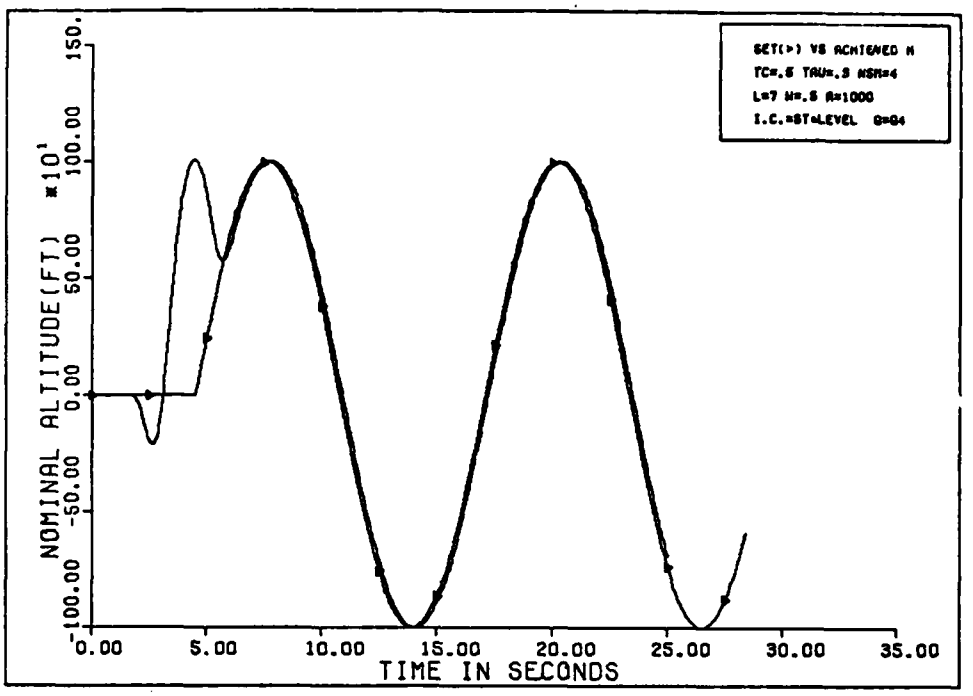


Fig 48. Set Altitude (Δ) and Achieved Altitude (Clean) for Q=Q4 and ω=0.5 Hz and TC=0.5 sec

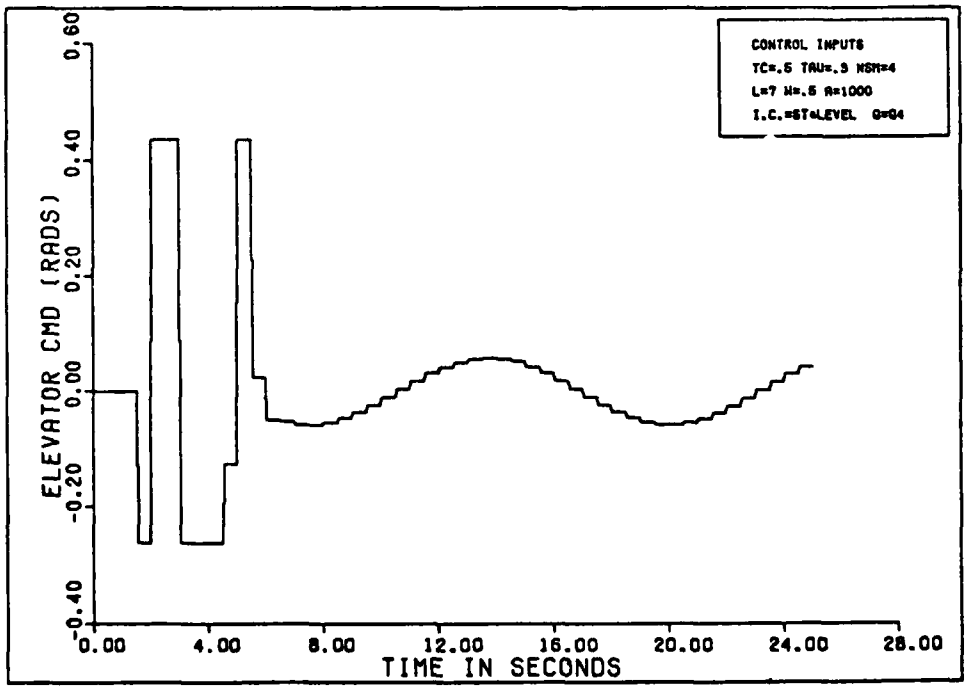


Fig 49. Control Inputs for Q=Q4 and ω=0.5 Hz and TC=0.5 sec

TABLE VI

Sinusoidal Frequencies Used and Peak and Average Altitude Errors for the Test Cases

Variable	Test Case 1	Test Case 2
A(ft)	1000	1000
ω_1(Hz)	0.5	0.5
ω_2(Hz)	0.6	0.6
ω_3(Hz)	0.2	0.8
Average Error (ft)	69.31	82.18
Peak Error (ft)	171.59	254.96

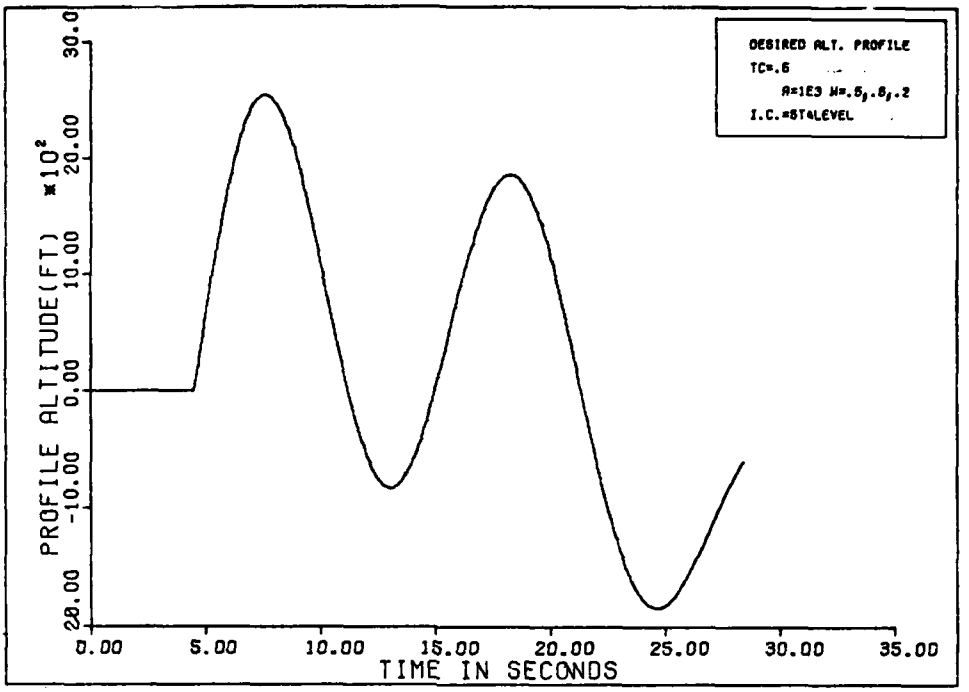


Fig 50. Set Path or Desired Altitude Profile for Test Case 1 (see Table VI)

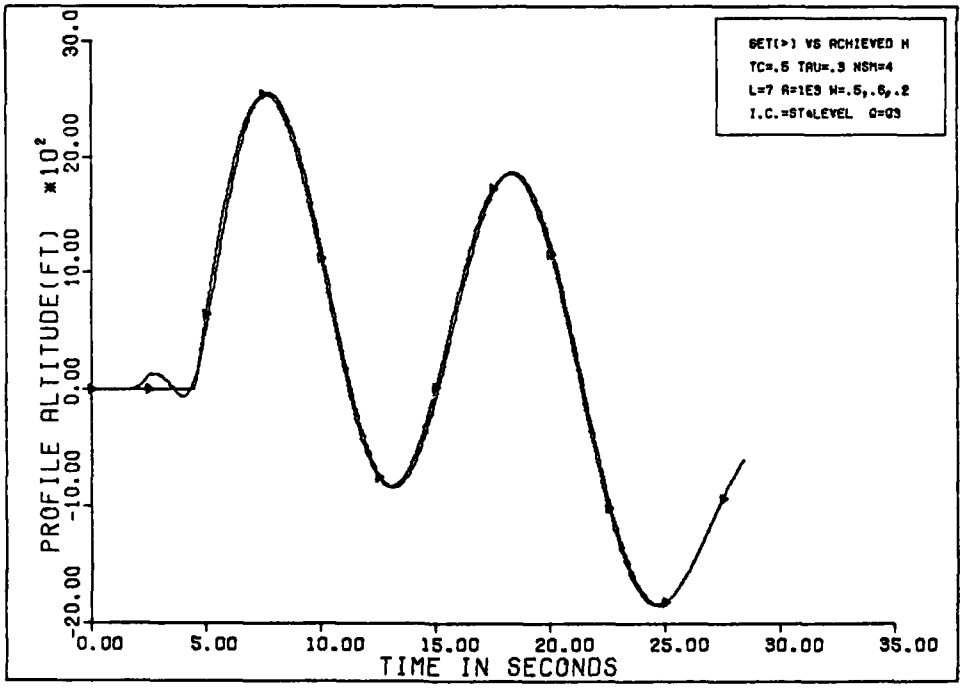


Fig 51. Desired Altitude (Δ) and Achieved Altitude (Clean) for Test Case 1 (see Table VI)

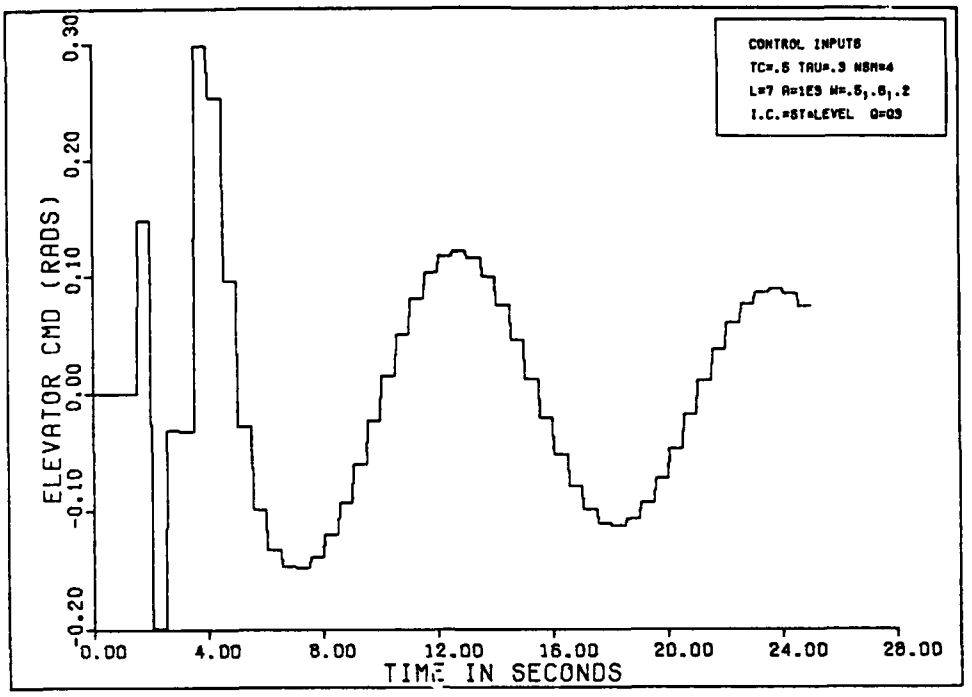


Fig 52. Control Inputs for Test Case 1 (see Table VI)

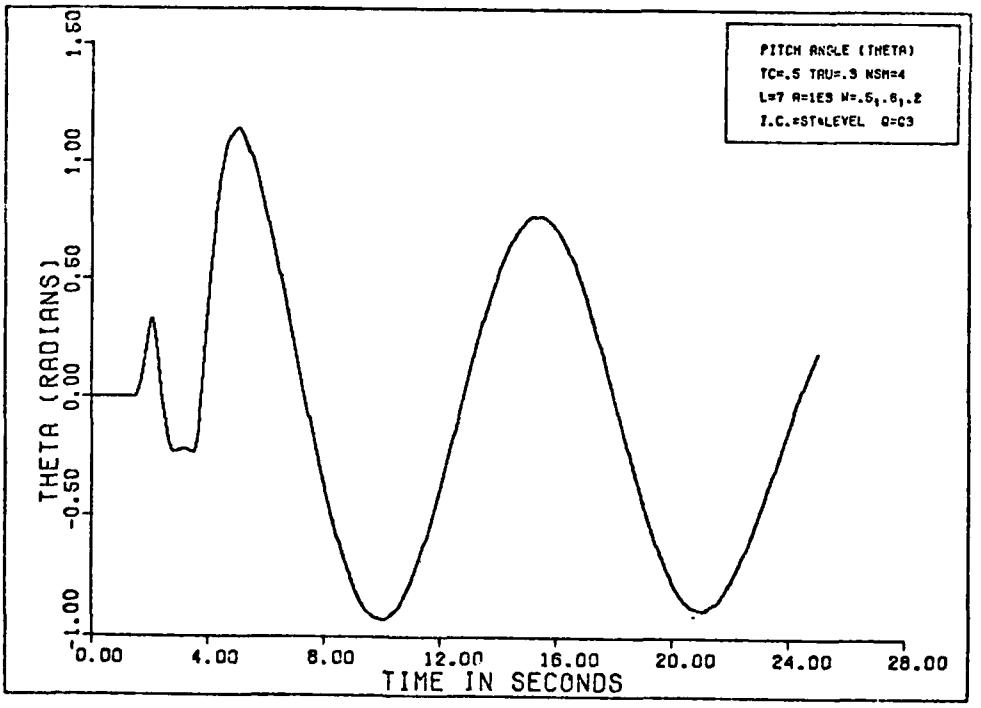


Fig 53. Pitch Angle (θ) for Test Case 1 (see Table VI)

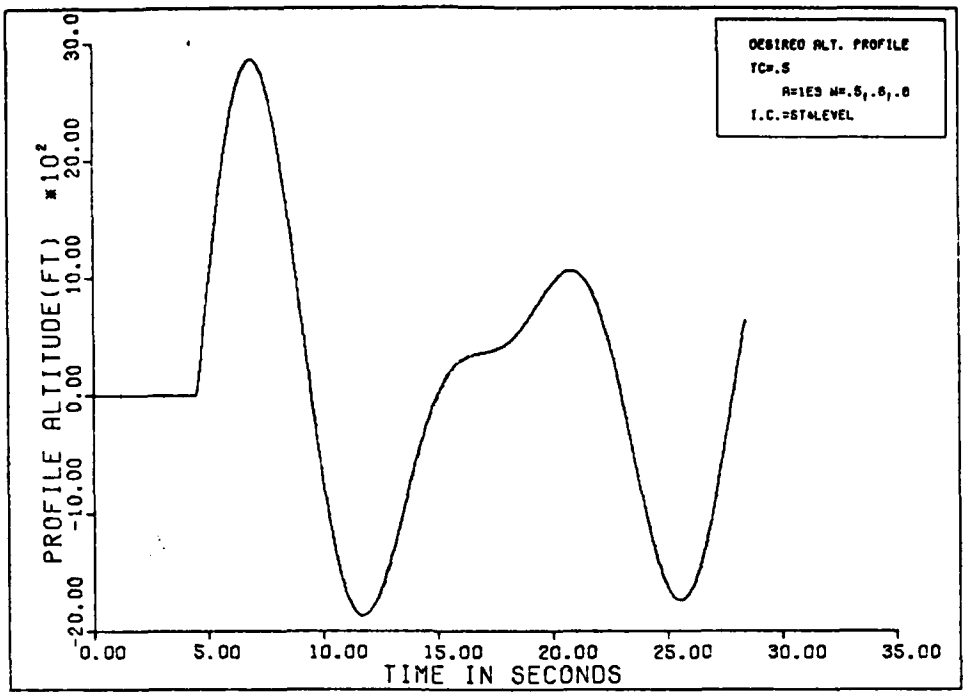


Fig 54. Set Path or Desired Altitude Profile for Test Case 2 (see Table VI)

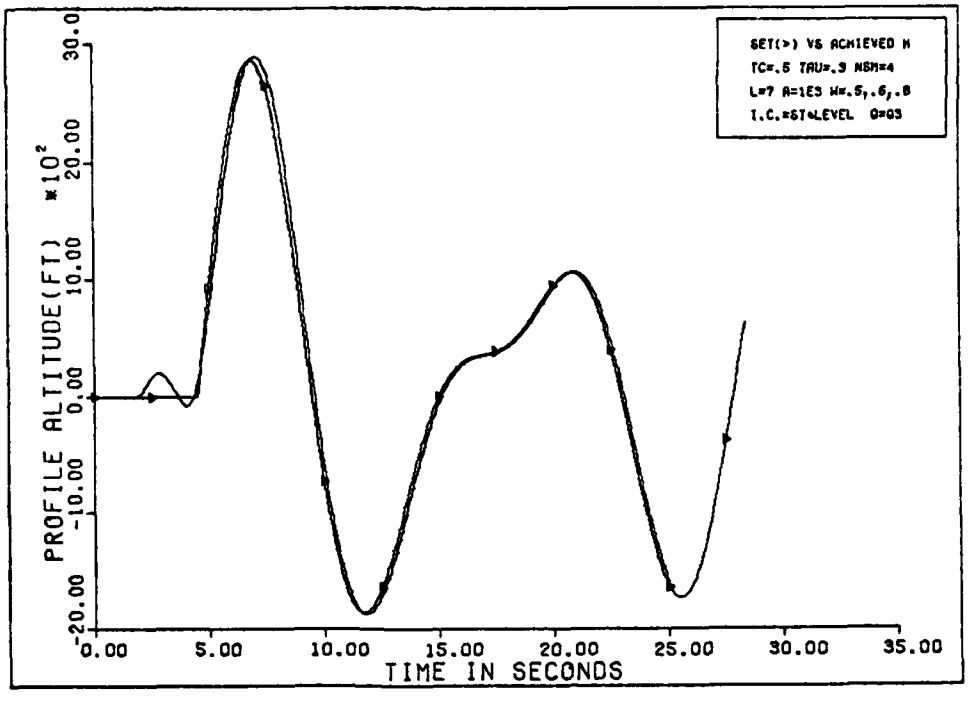


Fig 55. Desired Altitude (Δ) and Achieved Altitude (Clean) for Test Case 2 (see Table VI)

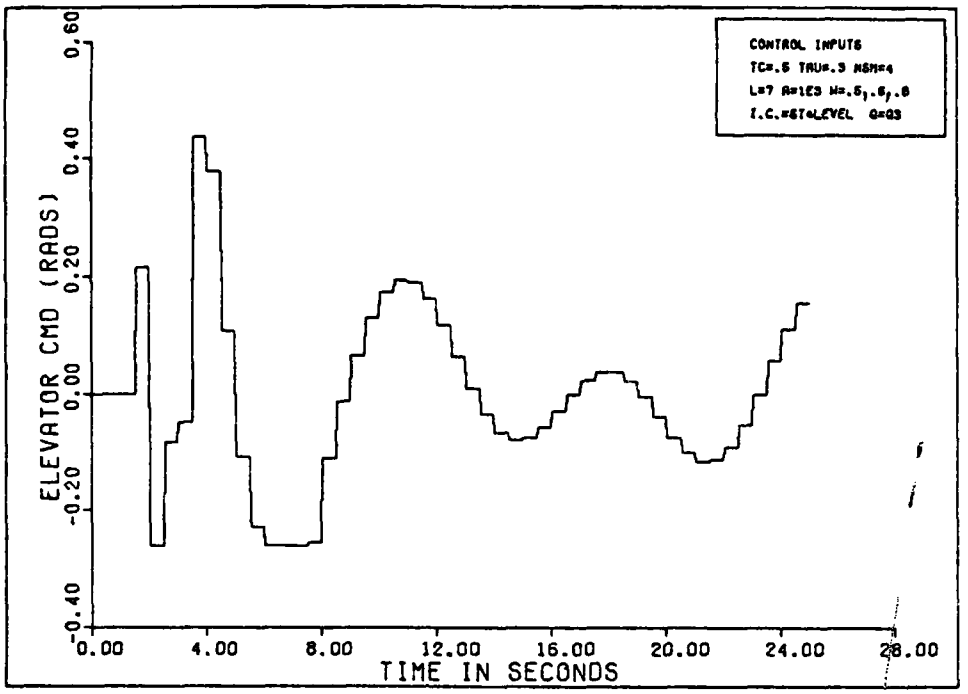


Fig 56. Control Inputs for Test Case 2 (see Table VI)

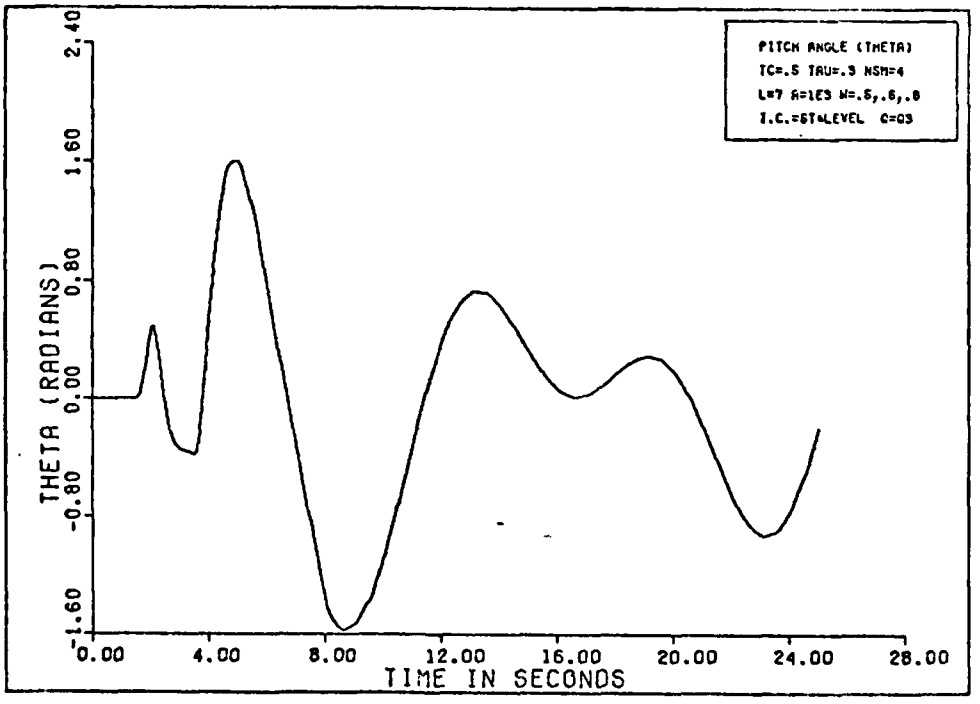


Fig 57. Pitch Angle (θ) for Test Case 2 (see Table VI)

angle for which the system model is valid. Although the altitude errors indicated in Figures 51 or 55 cannot be used to show that the controller/aircraft could actually fly the test profiles as indicated, the test results indicate that this controller algorithm is a viable one for the terrain following problem.

It is concluded that the algorithm is readily adaptable to a tracking problem. Limitations due to the linearized system model, however, make the results of the terrain following tests unusable except for comparison purposes. In order to get more meaningful results, the problem should be reformulated to run altitude as a function of down-range position and utilization of one of the following alternatives:

1. Go to on-line identification of the impulse function and use of a Kalman Filter for prediction purposes.

2. A series of linearized models can be stored, each to be used within the appropriate range of flight conditions (similar in concept to gain scheduling).

3. Make the controller so "robust" that the model mismatch has little effect.

Appendix D contains the FORTRAN code used for Test Case 1.

VI Conclusions and Recommendations

Chapter VI contains the conclusions drawn from this study of the smoothing approach to Output Predictive Control and some possible areas for further study.

Conclusions

In Ref 1, the control problem is formulated for exact matching (at control change time) of the system output to the desired trajectory and prediction of the zero input response is for one step ahead. For a system higher than second order the above formulation led to instability. This problem was corrected by looking further into the future for control calculation and by allowing some deviation of the output around the desired trajectory.

For the smoothing approach implemented as a regulator the results are good - even in the face of 30% model mismatch. The level of success or goodness, however, is dependent on the proper selection of the internal parameters of the regulator. The "best" set of parameters for the smoothing approach implementation approximates the Output Predictive Dead-Beat Controller (OPDEC) found in Refs 3 and 11. The most attractive feature of OPDEC is that there is only one internal parameter to be chosen, the control change time (TC), and a direct design procedure has been developed for the selection of TC .

Investigation of the terrain following application indicates that the algorithm is easily adapted to a tracking task. Limitations due to the linearized system model, however, make the current results

of the example inconclusive for a practical aircraft controller without the probable use of model adaption. However, this example illustrates good performance of the output predictive controller in a demanding tracking situation.

Recommendations for Further Study

Investigation in the following areas could provide further insights into the general area of Output Predictive Control:

1. Further development of the synthesis technique for selection of the internal parameters of the controller (i.e., TC , L , NSM , Q and Tau). A clearer understanding of the interrelationships of the parameters will facilitate application of the controller to new problems.

2. An indepth study of the robustness properties of the algorithm is necessary for quantification of these properties.

3. A reformulation of the tracking problem for more meaningful results would include running altitude as a function of downrange position and utilization of one of the following alternatives:

   a. Implementation of on-line identification of the impulse response function and use of a Kalman Filter for prediction.

   b. A series of linearized models can be stored, each to be used within the appropriate range of flight conditions (similar in concept to gain scheduling).

   c. Insure that the controller is so robust that the model mismatch introduced by using a linearized model outside its range has little effect on the closed loop performance.

Bibliography

1. Colson, H. J. Application of Model Algorithmic Control to a Lightly Damped Single Input Single Output System. MS Thesis. Wright-Patterson AFB OH: School of Engineering, Air Force Institute of Technology, December 1978.

2. Holtz, A. F. B-52E CCV Flutter Mode Control System Analysis Using Level 2.02 FLEXSTB. AFFDL Technical Memorandum TM-77-46-FGC. Wright-Patterson AFB OH: Air Force Flight Dynamics Laboratory, 1977.

3. Kirkwood, E. H. Robustness Studies of Output Predictive Dead-Beat Control for Wing Flutter Control Applications. MS Thesis. Wright-Patterson AFB OH: School of Engineering, Air Force Institute of Technology, December 1979.

4. Kuo, B. C. Digital Control Systems. Champaign IL: SRL Publishing Company, 1977.

5. Lawson, C. L. and R. J. Hanson. Solving Least Squares Problems. Englewood Cliffs NJ: Prentice-Hall, Inc., 1974.

6. Mehra, R. K. et al. "Model Algorithmic Control Using IDCOM for the F-100 Jet Engine Multivariable Control Design Problem," International Forum on Alternatives for Multivariable Control, 1977.

7. _____. "Model Algorithmic Control: Theoretical Results on Robustness," Proceedings 1979 Joint Automatic Control Conference: 387-392. New York: American Institute of Chemical Engineers, 1979.

8. Mereau, P. et al. "Flight Control Application of Model Algorithmic Control with IDCOM (Identification and Command)," Proceedings IEEE Conference on Decision and Control: 977-982 (1978).

9. Rao, C. R. Statistical Inference and Its Application. New York: John Wiley and Sons, 1973.

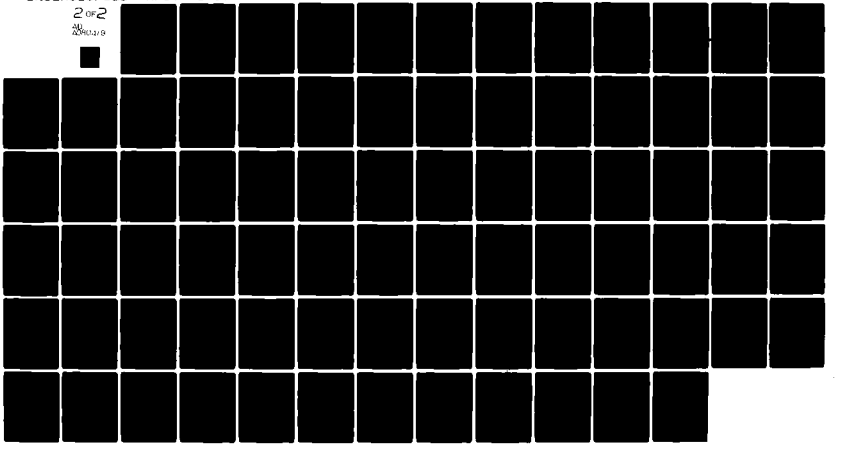
10. Reid, J. G. Lecture notes distributed in EE 5.10, "Linear Systems Analysis and Digital Computation Methods." School of Engineering, Air Force Institute of Technology, Wright-Patterson AFB OH, 1979.

11. _____. et al. "Robustness Properties of Output Predictive Dead-Beat Control: SISO Case," 1979 IEEE Decision and Control Conference, 1979.

12. Richalet, J. et al. "Model Predictive Heuristic Control: Application to Industrial Processes," Automatica, 14 (5):413 (1978).

AD-A080 419 AIR FORCE INST OF TECH WRIGHT-PATTERSON AFB OH SCHOO--ETC F/G 1/3
THE APPLICATION OF OUTPUT PREDICTIVE DIGITAL CONTROL TO WING FL--ETC(U)
DEC 79 D E CHAFFIN
UNCLASSIFIED AFIT/GE/EE/79-9 NL

2 OF 2
AD A080419



13. Rust, B. et al. "A Simple Algorithm for Computing the Generalized Inverse of a Matrix," Communications of the ACM, 9 (5):32-34 (1966).

Appendix A

Procedure for Solution of the Normal Equation

The procedure for solving the Normal Equation for a simple example problem is presented to illustrate the dimensions of its components and to show a series of matrix manipulations which result in a savings of computer storage space. Eq (28), the Normal Equation, is repeated for convenience:

$$\underline{\bar{u}} = (\bar{H}'Q\bar{H})^{-1}\bar{H}'Q\underline{z} \tag{28}$$

where

$\underline{\bar{u}}$ = An L dimension vector

$\bar{H}$ = An (NSM*L) X L matrix

Q = An (NSM*L) X (NSM*L) diagonal matrix

$\underline{z} = \underline{y}_d - \underline{y}_{zi}$ (21)

$\underline{z}$ = An (NSM*L) dimension vector

If three future control inputs per iteration are calculated (L=3), three smoothing terms per control input are used (NSM=3) and Q is chosen as Q3 , Eq (38), a weighting matrix discussed in Chapter III, then

$$\bar{H} = \begin{bmatrix} h(1) & 0 & 0 \\ h(1)+h(2) & 0 & 0 \\ h(1)+h(2)+h(3) & 0 & 0 \\ \hline \vdots & h(1) & 0 \\ \vdots & h(1)+h(2) & 0 \\ \sum_{i=1}^{6} h(i) & h(1)+h(2)+h(3) & 0 \\ \hline \vdots & \vdots & h(1) \\ \vdots & \vdots & h(1)+h(2) \\ \sum_{i=1}^{9} h(i) & \sum_{i=1}^{6} h(i) & h(1)+h(2)+h(3) \end{bmatrix} \quad \text{(A-1)}$$

$$Q = Q3 = \text{Diagonal}[1,2,4,8,16,32,..,256] \quad \text{(A-2)}$$

The following vectors and matrices are defined:

$$\underline{x}_1 = \begin{bmatrix} h(1) \\ h(1)+h(2) \\ h(1)+h(2)+h(3) \end{bmatrix} \quad \text{(A-3)}$$

$$\underline{x}_2 = \begin{bmatrix} h(1)+h(2)+..+h(4) \\ h(1)+h(2)+..+h(5) \\ h(1)+h(2)+..+h(6) \end{bmatrix} \quad \text{(A-4)}$$

$$\underline{x}_3 = \underline{x}_L = \begin{bmatrix} h(1)+h(2)+..+h(7) \\ h(1)+h(2)+..+h(8) \\ h(1)+h(2)+..+h(9) \end{bmatrix} \quad \text{(A-5)}$$

$$Q_1 = \text{Diagonal}[\ 1\ 2\ 4\ ] \quad \text{(A-6)}$$

$$Q_2 = \text{Diagonal}[\ 8\ \ 16\ \ 32\ ] \quad \text{(A-7)}$$

$$Q_3 = \text{Diagonal}[\ 64\ \ 128\ \ 256\ ] \quad \text{(A-8)}$$

and a piece of the Normal Equation can be written as

$$\bar{H}'Q\bar{H} = \begin{bmatrix} \underline{x}_1' & \underline{x}_2' & \underline{x}_3' \\ \underline{0}' & \underline{x}_1' & \underline{x}_2' \\ \underline{0}' & \underline{0}' & \underline{x}_3' \end{bmatrix} \begin{bmatrix} Q_1 & 0^* & 0^* \\ 0^* & Q_2 & 0^* \\ 0^* & 0^* & Q_3 \end{bmatrix} \begin{bmatrix} \underline{x}_1 & \underline{0} & \underline{0} \\ \underline{x}_2 & \underline{x}_1 & \underline{0} \\ \underline{x}_3 & \underline{x}_2 & \underline{x}_1 \end{bmatrix} \tag{A-9}$$

where

$$\underline{0} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{A-10}$$

$$0^* = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{A-11}$$

Carrying out the matrix multiplications indicated in Eq (A-9) and putting the result into a summation form yields

$$\bar{H}'Q\bar{H} = \left[\begin{array}{c|c|c} \sum_{k=1}^{3} \underline{x}_k' Q_k \underline{x}_k & \sum_{k=2}^{3} \underline{x}_k' Q_k \underline{x}_{k-1} & \sum_{k=3}^{3} \underline{x}_k' Q_k \underline{x}_{k-2} \\ \hline \sum_{k=2}^{3} \underline{x}_{k-1}' Q_k \underline{x}_k & \sum_{k=2}^{3} \underline{x}_{k-1}' Q_k \underline{x}_{k-1} & \sum_{k=3}^{3} \underline{x}_{k-1}' Q_k \underline{x}_{k-2} \\ \hline \sum_{k=3}^{3} \underline{x}_{k-2}' Q_k \underline{x}_k & \sum_{k=3}^{3} \underline{x}_{k-2}' Q_k \underline{x}_{k-1} & \sum_{k=3}^{3} \underline{x}_{k-2}' Q_k \underline{x}_{k-2} \end{array}\right] \tag{A-12}$$

which is symmetric.

Equivalent matrices, containing all of the information found in the original $\bar{H}$, Eq (22), and Q matrices can be formulated as

$$HEQ = [\underline{x}_1, \underline{x}_2, .., \underline{x}_L]_{NSMXL} \tag{A-13}$$

$$QEQ = [\underline{q}_1, \underline{q}_2, .., \underline{q}_L]_{NSMXL} \tag{A-14}$$

where, for the example problem $\underline{x}_1, \underline{x}_2$ and $\underline{x}_3 = \underline{x}_L$ are given by Eqs (A-3) through (A-5), and

$$\underline{q}_1 = \begin{bmatrix} 1 \\ 2 \\ 4 \end{bmatrix} \tag{A-15}$$

$$\underline{q}_2 = \begin{bmatrix} 8 \\ 16 \\ 32 \end{bmatrix} \tag{A-16}$$

$$\underline{q}_3 = \underline{q}_L = \begin{bmatrix} 64 \\ 128 \\ 256 \end{bmatrix} \tag{A-17}$$

Using these equivalent matrices (HEQ and QEQ) an inner product relationship can be used to produce $[\bar{H}'Q\bar{H}]$ instead of the straightforward transpose and matrix multiplications of Eqs (A-1) and (A-2). The first column of the original $\bar{H}$ matrix is formed using the relationship

$$h(i) = CF^{i-1}G \tag{10}$$

These elements are put into HFC , an (NSM*L) single dimension array. HEQ is dimensioned as an (NSMXL) matrix and equivalenced to HFC .

The FORTRAN code used to find $[\bar{H}'Q\bar{H}]$ using HEQ and QEQ is in Appendix B within Subroutine HBAR.

At this point the inverse of $[\bar{H}'Q\bar{H}]$ is required. The $[\bar{H}'Q\bar{H}]$ matrix may be less than full rank, requiring a generalized inverse. A library subroutine, GMINV, computes the generalized inverse through a process of column orthogonalization using a Gram-Schmidt procedure (Ref 13).

ZEQ , an (NSMXL) matrix is equivalenced to Z , an (NSM*L) single dimension array containing the elements found using Eq (21).

$$ZEQ = [\underline{z}_1, \underline{z}_2, .., \underline{z}_L]_{NSMXL} \tag{A-18}$$

where, for the example problem

$$\underline{z}_1 = \begin{bmatrix} y_d(1) - y_{zi}(1) \\ y_d(2) - y_{zi}(2) \\ y_d(3) - y_{zi}(3) \end{bmatrix} \tag{A-19}$$

$$\underline{z}_2 = \begin{bmatrix} y_d(4) - y_{zi}(4) \\ y_d(5) - y_{zi}(5) \\ y_d(6) - y_{zi}(6) \end{bmatrix} \tag{A-20}$$

$$\underline{z}_3 = \underline{z}_L = \begin{bmatrix} y_d(7) - y_{zi}(7) \\ y_d(8) - y_{zi}(8) \\ y_d(9) - y_{zi}(9) \end{bmatrix} \tag{A-21}$$

An expression similar to Eq (A-12) can be built and an inner product relationship similar to the one used to find $[\bar{H}'Q\bar{H}]$ can be used to calculate $[\bar{H}'Q\underline{z}]$. The FORTRAN code used to find $[\bar{H}'Q\underline{z}]$ using HEQ , QEQ and ZEQ is in Appendix B within Subroutine CONTROL.

As an indication of the savi .gs realized when using the HEQ , QEQ and ZEQ matrices versus $\bar{H}$, Q and $\underline{z}$, consider the solution of the Normal Equation for twenty step ahead prediction (L=20) and ten smoothing terms (NSM=10). Matrix dimensions are

$$[\bar{H}]_{200X10} \quad \text{versus} \quad [HEQ]_{20X10}$$

$$[Q]_{200X200} \quad \text{versus} \quad [QEQ]_{20X10}$$

Core memory requirements (in base 10) for these two matrices alone are 42000 words required for $\bar{H}$ and Q versus 400 words using HEQ and QEQ .

Appendix B

Sample FORTRAN Code Used in the Investigation of the Algorithm Implemented as a Regulator

The basic program in this Appendix follows the functional block diagram of Figure 1. Addition of a set of logical switches allows investigation of model mismatch (Chapter IV) and the effects of noisy measurements and inputs. Comment statements at the front of the program explain the use of these switches.

The only external to the program is a subroutine used to find the generalized inverse of a matrix (GMINV). This subroutine is on a functional library named CONTROL, ID=L 720033, SN=AFML. CONTROL must be attached and libraried before compilation of the program.

The output of this particular version of the algorithm is Figures 28 through 31 in Chapter III.

```
      PROGRAM REGULAT(INPUT,OUTPUT,PLOT,TAPE5=INPUT,                    000100
     1TAPE6=OUTPUT)                                                     000110
C***********************************************************************000120
C                                                                       000130
C  THIS PROGRAM HAS BEEN CODED TO HANDLE A SISO SYSTEM ONLY.            000140
C  VARIABLES OF INTEREST INCLUDE AN OVERALL CONTROL TIME,TC,             000150
C  WHICH IS HOW OFTEN THE CONTROL INPUT IS CHANGED.                      000160
C                                                                       000170
C  NSMOOTH(NSM) IS THE NUMBER OF DESIRED OUTPUT POINTS USED             000180
C  TO SOLVE A LEAST SQUARES CURVE FIT PROBLEM IN AN ATTEMPT              000190
C  TO LIMIT OSCILLATIONS OF THE OUTPUT.                                 000200
C                                                                       000210
C  TDESIRE=TC/NSMOOTH IS THE SAMPLE TIME AT WHICH THE DESIRED           000220
C  POINTS ON A DESIRED TRAJECTORY TO A SET POINT ARE CALCULATED.        000230
C                                                                       000240
C  AFTER THE TRAJECTORY( WITH TIME CONSTANT TAU) IS CALCULATED, L       000250
C  INPUTS ARE CALCULATED(ONLY THE FIRST OF WHICH IS APPLIED)            000260
C  TO DRIVE THE OUTPUT TO FOLLOW THE DESIRED TRAJECTORY.                000270
C                                                                       000280
C  THE OUTPUT IS SAMPLED AFTER APPLICATION OF THE CONTROL TO            000290
C  KEEP TABS ON THE OSCILLATIONS. SAMPLE TIME OF THE OUTPUT             000300
C  IS TOBS=TDESIRE/NOBS=TC/NSM*NOBS. NOBS NSM IS HOW MANY               000310
C  TIMES THE OUTPUT IS CHECKED BETWEEN CONTROL CHANGES.                 000320
C                                                                       000330
C                                                                       000340
C  WHEN CHANGING N,NSM,L,NOBS MUST ALSO CHANGE DIMENSION                000350
C  STATEMENTS.  ML=NSM*L IS A MATRIX SIZE USED IN THE NORMAL            000360
C  EQUATIONS.  NDR1,NDR2 AND NDR3 ARE PARAMETERS USED TO                000370
C  "VARIABLE DIMENSION" THE ARRAYS WITHIN THE SUBROUTINES.              000380
C  IF THE DIMENSIONS IN THE MAIN PROGRAM ARE CHANGED, WILL              000390
C  ONLY HAVE TO CHANGE THE DIMENSION BLOCK WITHIN THE                   000400
C  MAIN PROGRAM AND THE PARAMETERS NDR1,NDR2 ANDR3.                     000410
C  YOU WILL NOT HAVE TO CHANGE THE DIMENSION STATEMENTS WITHIN          000420
C  THE SUBROUTINES. NDR1=N, NDR2=ML=NSM*L, NDR3=L.                      000430
C                                                                       000440
C  DON'T FORGET TO CHANGE THE PLOT LABELS(IDYS XIDUS) TO                000450
C  REFLECT ANY CHANGED INFO.                                            000460
C                                                                       000470
C  WHEN THE NUMBER OF CONTROL INPUTS APPLIED (KF),NSM OR NOBS           000480
C  CHANGES, MUST ALSO CHANGE THE DIMENSIONS OF                          000490
```

```
C  YS,US,TSS TO KF*NOBS*NSM+3. ALSO REFLECT THIS CHANGE                  000500
C  IN THE NUMBER OF POINTS TO BE PLOTTED IN THE                          000510
C  CALL TO THE PLOTTING ROUTINES AND DATA PRINTOUT STATE-                000520
C  MENT. INPUT NTP(# TERMS TO BE PRINTED/PLOTTED)=KF*NOBS*NSM+1          000530
C                                                                        000540
C   NOTE** WHEN GO TO AN ITERATION OF L/TAU, WILL HAVE TO                000550
C   RE-INITIALIZE THE B AND X MATRICES.  THEY ARE DESTROYED              000560
C   WITHIN THE PROGRAM.                                                  000570
C                                                                        000580
C  IN THIS CODING, H,Q,AND Z WILL BE REPLACED BY EQUIVALENT              000590
C  MATRICES HFQ,QEQ,AND ZEQ FOR COMPUTATIONAL EFFICIENCY.                000600
C  BECAUSE OF THE NATURE OF THIS CODING, QC, HFC, AND ZC                 000610
C  MUST BE DIMENSIONED EXACTLY (ML=NSM*L).                               000620
C  QEQ, ZEQ, AND HEQ MUST ALL BE DIMENSIONED EXACTLY (NSM,L).            000630
C                                                                        000640
C                                                                        000650
C                                                                        000660
C                                                                        000670
C   SIZES OF THE ARRAYS ARE AS FOLLOWS:                                  000680
C   COM1(BIGGEST SQUARE ARRAY PASSED TO CLYMAN ROUTINES)                 000690
C   A(N,N),B(N),C(N),X(N),XE(N),FS(N,N),GS(N),FL(N,N)                    000700
C   GL(N),EPINT(N,N),CFL(N),CFLK(N),HTQH(L,L)                           000710
C   HTQHI(L,L),YQ(ML),YZ(ML),ZC(ML),ZEQ(NSM,L)                          000720
C   UCOMP(L),HTQZ(L),GSU(N),FSX(N),HFC(ML),HEQ(NSM,L)                    000730
C   QC(ML),QEQ(NSM,L),YS,US,TSS(NSM*NOBS*KF+3)                          000740
C   ADYS(IOJS(17)                                                        000750
C                                                                        000760
C  AT,BT,CT,GLT,ETC ALL REFER TO THE MATRIX DESCRIPTION OF               000770
C  THE "TRUTH MODEL" SYSTEM.                                             000780
C  AM,BM,CM,GLM,ETC ALL REFER TO THE MATRIX DESCRIPTION OF               000790
C  THE "ACTUAL" OR MODEL SYSTEM.                                         000800
C                                                                        000810
C  THE OPTION OF INPUTTING RANDOM INITIAL STATES HAS                     000820
C  BEEN PROVIDED.                                                        000830
C                                                                        000840
C  FOUR "SWITCH" SETINGS HAVE BEEN INCORPORATED FOR                      000850
C  ADDING "NOISE" SOURCES TO THE SYSTEM.                                 000860
C  FIRST SWITCH>>>> IF SW1=.TRUE.,THE TRUTH MODEL MATRICES               000870

C  FIRST SWITCH>>>> IF SW1=.TRUE.,THE TRUTH MODEL MATRICES               000870
```

```
C  FIRST SWITCH>>>> IF SW1=.TRUE.,THE TRUTH MODEL MATRICES          0.0870
C  WILL BE SENT TO SOLVE FOR THE ZERO INPUT RESPONSE.              0.0880
C  IF SW1=.FALSE., THE ACTUAL OR MODEL SYSTEM WILL BE USED.        0.0890
C  THIS WILL INTRODUCE A "NOISE" INTO THE SYSTEM IN THAT           0.0900
C  THE ZERO INPUT REPONSE WILL BE PREDICTED USING A SYSTEM         0.0910
C  MODEL DIFFERENT FROM THE ONE UPON WHICH THE CONTROLLER          0.0920

C  MODEL DIFFERENT FROM THE ONE UPON WHICH THE CONTROLLER          0.0920
C  IS BASED.                                                       0.0930
C                                                                  0.0940
C  SECOND SWITCH>>>> IF SW2=.TRUE., GT AND GLT WILL BE USED        0.0950
C  TO FIND THE IMPULSE FUNCTION WHICH WILL BE EVENTUALLY           0.0960
C  SENT TO SUB CONTROL TO PREDICT FUTURE INPUTS NEEDED TO          0.0970
C  FOLLOW THE DESIRED TRAJECTORY.                                  0.0980
C  IF SW2=.FALSE., GM AND GLM WILL BE USED IN THIS PREDICTIVE      0.0990
C  FUNCTION.                                                       0.1000
C                                                                  0.1010
C  IF THE POSSIBILITY OF MODEL MISMATCH IS NOT TO BE               0.1020
C  EXPLORED, THE SAME MATRICES SHOULD BE USED FOR AM,              0.1030
C  BM,CM AS WAS USED FOR AT,BT,CT. EITHER SETTING FOR              0.1040
C  SW1 AND SW2 WILL RESULT IN THE SAME INFO FROM                   0.1050
C  HOAR AND YPI.                                                   0.1060
C                                                                  0.1070
C  THIRD SWITCH>>>> IF SW3=.TRUE.,RANDOM NOISE WILL BE             0.1080
C  USED TO DRIVE THE SYSTEM(IN ADDITION TO THE DETERMINISTIC       0.1090
C  INPUT).  THE STATE UPDATE EQUATION WILL BE:                     0.1100
C  X(K+1)=F*X(K)+G*(U+W), WHERE W IS DISCRETE WHITE GAUSSIAN       0.1110
C  NOISE OF SELECTABLE STRENGTH.                                   0.1120
C                                                                  0.1130
C  FOURTH SWITCH>>>> IF SW4=.TRUE. AN ADDITIONAL NOISE             0.1140
C  WILL BE ADDED TO THE MEASUREMENT OF Y, RESULTING IN             0.1150
C  Y=C*X+V.                                                        0.1160
C  IF SW4=.FALSE., NO MEASUREMENT NOISE IS ADDED.                  0.1170
C                                                                  0.1180
C  THE SUBROUTINES DSCRT,GMINV AND DSP ARE IN A                    0.1190
C  PERSONAL LIBRARY. I STOLE DSCRT AND GMINV FROM                  0.1200
C  CONTROL,ID=L720033,SN=AFML. DSP IS ON THE AFIT                  0.1210
C  SUBROUTINES.                                                    0.1220
C                                                                  0.1230
```

```
C                                                                       0001240
C*****                                                                  0001250
C                                                                       0001260
C                                                                       0001270
      LOGICAL SW1,SW2,SW3,SW4                                           0001280
      DIMENSION CCM1(7,7),AT(4,4),BT(4),CT(4),X(4)                      0001290
      DIMENSION AM(4,4),BM(4),CM(4),EAINT(4,4)                          0001300
      DIMENSION FLT(4,4),GLT(4),FST(4,4),GST(4)                         0001310
      DIMENSION FLM(4,4),GLM(4),FSM(4,4),GSM(4)                         0001320
      DIMENSION CFL(4),CFLK(4),HFC(28),HEC(4,7)                         0001330
      DIMENSION QC(28),QEC(4,7),FGX(4),WK4(28)                          0001340
      DIMENSION HFCH(7,7),HFCHI(7,7),HICZ(7),UCOMP(7),GSU(4)            0001350
      DIMENSION YC(28),YZ(28),ZC(28),ZEC(4,7)                           0001360
      DIMENSION YS1(403),YS2(403),YS3(403),YST(1205),IDYS(17)           0001370
      DIMENSION US1(403),US2(403),US3(403),UST(1205),IDUS(17)           0001380
      DIMENSION TSS1(403),TSS2(403),TSS3(403)                           0001390
      COMMON/MAIN1/NDIM,NDIM1,COM1/INOU/KIN,KOUT,KPUNCH                 0001400
C                                                                       0001410
C                                                                       0001420
      EQUIVALENCE (HFC(1),HEC(1,1)),(QC(1),QEC(1,1))                    0001430
      EQUIVALENCE (ZC(1),ZEC(1,1))                                      0001440
      EQUIVALENCE (YS1(1),YST(1)),(YS2(1),YST(402))                     0001450
      EQUIVALENCE (US1(1),UST(1)),(US2(1),UST(402))                     0001460
      EQUIVALENCE (YS3(1),YST(803)),(US3(1),UST(803))                   0001470
C  THE SYSTEM EQNS ARE IN PHASE VBL FORM, B(N) IS                       0001480
C  REALLY "1", THE ENTRY IN ITS PLACE IS A SCALE                        0001490
C  FACTOR FOR THE INPUT. THIS DOES NOT AFFECT THE                       0001500
C  OUTPUT.                                                              0001510
C                                                                       0001520
      DATA (AT(I,1),I=1,4)/3*0.,-8511.7698E/                            0001530
      DATA (AT(I,2),I=1,4)/1.0,2*0.,-279.098/                           0001540
      DATA (AT(I,3),I=1,4)/0.,1.,0.,-274.975/                           0001550
      DATA (AT(I,4),I=1,4)/2*0.,1.,-1.6/                                0001560
      DATA (AM(I,1),I=1,4)/3*0.,-.2460E5/                               0001570
      DATA (AM(I,2),I=1,4)/1.0,2*0.,-613.2/                             0001580
      DATA (AM(I,3),I=1,4)/0.,1.,0.,-183.2/                             0001590
      DATA (AM(I,4),I=1,4)/0.,1.,0.,-183.2/                             0001600
      DATA (AM(I,4),I=1,4)/2*0.,1.,-2.08/                               0001610
      DATA (BM(I),I=1,4)/3*0.,.2460E5/                                  0001620
```

```
      DATA (CM(I),I=1,4)/1.3,3*0./                                      0.1630
      DATA BT/3*0.,8611.769305/                                         001640
      DATA CT/1.0,3*0./                                                 001650
C  NDR1=AT LEAST N,NDR2=AT LEAST ML,NDR3=AT LEAST L                     001660
      DATA YSET,NT,N,NDR1,NDR2/0.,30,4,4,28/                            001670

      DATA YSET,NT,N,NDR1,NDR2/0.,30,4,4,28/                            001670

      DATA YSET,NT,N,NDR1,NDR2/0.,30,4,4,28/                            001670
      DATA TC,L,NDR3,KF,NDBE,NSM/.075,7,7,20,5,4/                       001690
      DATA TAU,NTP/0.1,4001/                                            001690
      DATA IDYS(1)/20HTC=0.075 CLEAN      /                             001700
      DATA IDYS(3)/20H> MARKS TC=0.125    /                             001710
      DATA IDYS(5)/20H0 MARKS TC=1.200    /                             001720
      DATA IDYS(7)/20HNSM=4 L=7 Q=05      /                             001730
      DATA IDYS(9)/20H   TIME IN SECONDS  /                             001740
      DATA IDYS(11)/20H   SAMPLED OUTPUT  /                             001750
      DATA IDYS(13)/50HFIG  . OUTPUT VARIATIONS FOR VARIABLE TC         001760
     1 /                                                                001770
C                                                                       001780
C                                                                       001790
      DATA IDUS(15)/20H   CONTROL INPUT   /                             001800
      DATA IDUS(17)/50HFIG  . CONTROL VARIATIONS FOR VARIABLE TC        001810
     1 /                                                                001820
      KK<=3                                                             001830
      IPASS=1                                                           001840
      NTPT=1203                                                         001850
      SW1=.TRUE.                                                        001860
      SW2=.TRUE.                                                        001870
      SW3=.FALSE.                                                       001880
      SW4=.FALSE.                                                       001890
30    IPASS=IPASS+1                                                     001900
      IF(IPASS.EQ.2)TC=.124                                             001910
      IF(IPASS.EQ.3)TC=.200                                             001920
      IF(IPASS.GE.4)GO TO 130                                           001930
C                                                                       001940
C                                                                       001950
      DO 45 I=1,1                                                       001980
        IDUS(I)=IDYS(I)                                                 001990
45    CONTINUE                                                          002000
```

```
C  REINITIALIZE X AND B                                                  002010
      BT(1)=0.                                                          002020
      BT(2)=0.                                                          002030
      BT(3)=0.                                                          002040
      BT(4)=8511.-69806                                                 002050
      BM(1)=0.                                                          002060
      BM(2)=0.                                                          002070
      BM(3)=0.                                                          002080
      BM(4)=.246065                                                     002090
C                                                                       002100
C  THIS WILL GIVE THE SAME INITIAL STATE VECTOR FOR ALL                 002110
C  COMBINATIONS OF SWITCH SETTINGS.                                     002120
      CALL RANSET(3)                                                    002130
C                                                                       002140
C                                                                       002150
C  HERE IS WHERE CAN READ IN RANDOM INITIAL STATES.                     002160
C  THESE WILL BE READ IN AND REINITIALIZED                              002170
C  AS FOLLOWS                                                           002180
      X(1)=20.                                                          002190
      X(2)=10.                                                          002200
      X(3)=5.                                                           002210
      X(4)=0.                                                           002220
      PRINT*,"INITIAL STATE"                                            002230
      CALL PRINTR(N,X,NOR1)                                             002240
C                                                                       002250
      PRINT*," "                                                        002260
      PRINT*,"SWITCH SETTINGS ARE AS FOLLOWS:"                          002270
      WRITE(6,50)(IDYS(I),I=1,8)                                        002280
50    FORMAT((1H ,2A10,/))                                              002290
C                                                                       002300
      MN=NSM*NOBS                                                       002310
      TDESIRE=TC/NSM                                                    002320
      TOBS=TC/MN                                                        002330
      ALPHA=EXP(-TDESIRE/TAU)                                           002340
      ML=NSM-1                                                          002350
      KS=1                                                              002360
C  FORM Y(J)=CMX(0)                                                     002370
      Y=0.0                                                             002380
      DO 57 K=1,N                                                       002390
        Y=Y+CM(K)*X(K)                                                  002400
```

```
      YS1(1)=Y                                                          0C2410
      YS2(1)=Y                                                          0C2415
      YS3(1)=Y                                                          0C2420
57    CONTINUE                                                          0C2430
      US1(1)=0.                                                         0C2435
      US2(1)=0.                                                         0C2440
      US3(1)=0.                                                         0C2445
      TSS1(1)=0.                                                        0C2450
      TSS2(1)=0.                                                        0C2455
      TSS3(1)=0.                                                        0C2460
      KI4=5                                                             0C2470
      KOUT=6                                                            0C2480
      KPUNCH=7                                                          0C2490
C                                                                       0C2500
      PRINT ,"FOR THIS RUN TCONTROL=",TC,",TDESIRED=",TDESIRE           0C2510
      PRINT ," "                                                        0C2520
      PRINT ,"TOBSERVE=",TOBS,",TAU=",TAU,",ALPHA=",ALPHA               0C2530
      PRINT ," "                                                        0C2540
      PRINT ,"NSMOOTH(NSM)=",NSM,",NOBS=",NOBS,",L=",L                  0C2550
C                                                                       0C2560
C                                                                       0C2570
      CALL DISCRET(AT,BT,AM,BM,TDESIRE,TOBS,N,NT,NDR1,                  0C2580
     +DAINT,FLT,GLT,FST,GST,FL1,GLM,FSM,GSM)                            0C2590
C                                                                       0C2600
      CALL WEIGHT(NSM,L,ML,DC,DEQ)                                      0C2610
      IF(SW2)GO TO 61                                                   0C2620
      CALL HDAR(CM,FLM,GFL,DFLK,GLM,HFC,HEQ,QEQ,HTQH,HTQHI,             0C2630
     +N,L,ML,NSM,NDR1)                                                  0C2640
      GO TO 62                                                          0C2650
61    CONTINUE                                                          0C2660
      CALL HDAR(CT,FLT,GFL,DFLK,GLT,HFC,HEQ,QEQ,HTQH,HTQHI,             0C2670
     +N,L,ML,NSM,NDR1)                                                  0C2680
62    CONTINUE                                                          0C2690
C                                                                       0C2700
C  FINALLY THROUGH WITH THE PRELIMINARIES,CAN NOW                       0C2710
C  GET DOWN TO BUSINESS OF FINDING AN APPROPIATE CONTROL                0C2720
C  BY LOOKING AT THE DESIRED PATH,THE ZERO INPUT RESPONSE               0C2730
C  AND THE SET POINT.                                                   0C2740
C                                                                       0C2750
      DO 192 KC=1,KF                                                    0C2760
```

```
C                                                                       002770
C  REMOVED HEADERS FOR DESIRED OUTPUT POINTS                            002780
C                                                                       002790
      CALL YDESIRE(Y,YSET,A_PHA,ML,YD)                                  002800
C                                                                       002810
C  SUBROUTINE YZI DESTROYS THE C MATRIX.  COPY C MATRIX INTO            002820
C  THE B MATRIX .                                                       002830
      DO 75 I=1,N                                                       002840
        BT(I)=CT(I)                                                     002850
        BM(I)=CM(I)                                                     002860
75    CONTINUE                                                          002870
C                                                                       002880
C                                                                       002890
      IF(SW1)GO TO 80                                                   002900
      CALL YZI(X,N,ML,YZ,BM,FLM,CFL,NOR1)                               002910
      GO TO 90                                                          002920
C                                                                       002930
80    CONTINUE                                                          002940
      CALL YZI(X,N,ML,YZ,BT,FLT,CFL,NOR1)                               002950
90    CONTINUE                                                          002960
C                                                                       002970
      CALL CONTROL(YD,YZ,ZC,ZEQ,HEQ,DEQ,HTQHI,HTQZ,UCOMP,              002980
     +U,L,ML,NSM,NOR3)                                                  002990
C                                                                       003000
C  NOW UPDATE STATES AND SAMPLE THE OUTPUT OF THE "ACTUAL" SYSTEM.      003010
C                                                                       003020
C                                                                       003030
      DO 150 KO=1,MN                                                    003040
C                                                                       003050
      IF(.NOT.SW3) GO TO 100                                            003060
      CALL NOISE(1.0,0.,WN)                                             003070
      UN=U+WN                                                           003080
      CALL STATE(X,FST,FSX,GST,GSU,N,UN,NOR1)                           003090
      GO TO 105                                                         003100
100   CONTINUE                                                          003110
        CALL STATE(X,FST,FSX,GST,GSU,N,U,NOR1)                          003120
C                                                                       003130
105   CONTINUE                                                          003140
        CALL OUTPUT(X,CM,Y,N)                                           003150
      IF(.NOT.SW4) GO TO 111                                            003160
```

```
      CALL NOISE(1.0,0.,WN)                                              003170
      Y=Y+WN                                                             003180
110   CONTINUE                                                           003190
C                                                                        003200
        KS=KS+1                                                          003210
      IF(IPASS.EQ.1)YS1(KS)=Y                                            003240
      IF(IPASS.EQ.1)US1(KS)=U                                            003250
      IF(IPASS.EQ.1)TSS1(KS)=TSS1(KS-1)+TOBS                             003260
      IF(IPASS.EQ.2)YS2(KS)=Y                                            003270
      IF(IPASS.EQ.2)US2(KS)=U                                            003280
      IF(IPASS.EQ.2)TSS2(KS)=TSS2(KS-1)+TOBS                             003290
      IF(IPASS.EQ.3)YS3(KS)=Y                                            003300
      IF(IPASS.EQ.3)US3(KS)=U                                            003310
      IF(IPASS.EQ.3)TSS3(KS)=TSS3(KS-1)+TOBS                             003320
C                                                                        003330
150    CONTINUE                                                          003340
160    CONTINUE                                                          003350
      IF(IPASS.EQ.1.OR.IPASS.EQ.2)GO TO 30                               003360
      CALL PLOT(0.,-4.,-3)  ! CALL PLOT(0.,0.03,-3)                      003370
C                                                                        003380
C  UNLESS ONE TO (R KF, TIME FOR THE RUN IS THE SAME.                    003390
C  ONLY DIFFERENCE BETWEEN TSS1,TSS2,TSS3 IS THAT                        003400
C  SOME ARE DISPLAYED FINER THAN OTHERS.                                 003410
      CALL SCALE(TSS3,7.0,NTP,1)                                         003420
      PRINT*,"SCALED TSS3"                                               003430
      CALL SCALE(YST,3.,NTPT,1)                                          003440
      PRINT*,"SCALED YST"                                                003450
      YS21=YS2(1)   ;  YS31=YS3(1)                                       003460
      YS22=YS2(2)   ;  YS32=YS3(2)                                       003470
      YS1(NTP+1)=YST(NTPT+1)                                             003480
      YS1(NTP+2)=YST(NTPT+2)                                             003490
      TSS1(NTP+1)=TSS3(NTP+1)                                            003495
      TSS1(NTP+2)=TSS3(NTP+2)                                            003496
      CALL PGRAPH(TSS1,YS1,NTP,IDYS,-1,0,0)                              003500
      YS2(1)=YS21                                                        003510
      YS2(2)=YS22                                                        003520
      YS2(NTP+1)=YST(NTPT+1)                                             003530
      YS2(NTP+2)=YST(NTPT+2)                                             003540
      TSS2(NTP+1)=TSS3(NTP+1)                                            003550
      TSS2(NTP+2)=TSS3(NTP+2)                                            003560
```

```
      CALL MGRAPH(TSS2,YS2,NTP,IDYS,2,20,2)                          003570
      YS3(1)=YS31                                                    003580
      YS3(2)=YS32                                                    003590
      CALL MGRAPH(TSS3,YS3,NTP,IDYS,2,30,1)                          003600
C                                                                    003610
      CALL SCALE(TSS3,7.,NTP,1)                                      003620
      PRINT,"SCALED TSS3 FOR MGRAPH"                                 003630
      CALL SCALE(UST,5.,NTPT,1)                                      003640
      PRINT,"SCALED UST FOR MGRAPH"                                  003650
      US31=US3(1)   &  US21=US2(1)                                   003660
      US32=US3(2)   &  US22=US2(2)                                   003670
      TSS1(NTP+1)=TSS2(NTP+1)                                        003680
      TSS1(NTP+2)=TSS2(NTP+2)                                        003690
      US1(NTP+1)=UST(NTPT+1)                                         003691
      US1(NTP+2)=UST(NTPT+2)                                         003692
      CALL MGRAPH(TSS1,US1,NTP,IDUS,-1,0,3)                          003700
      US2(1)=US21                                                    003710
      US2(2)=US22                                                    003720
      US2(NTP+1)=UST(NTPT+1)                                         003730
      US2(NTP+2)=UST(NTPT+2)                                         003740
      TSS2(NTP+1)=TSS3(NTP+1)                                        003750
      TSS2(NTP+2)=TSS3(NTP+2)                                        003760
      CALL MGRAPH(TSS2,US2,NTP,IDUS,-1,20,2)                         003770
      US3(1)=US31                                                    003780
      US3(2)=US32                                                    003790
      CALL MGRAPH(TSS3,US3,NTP,IDUS,-1,30,1)                         003800
190   CONTINUE                                                       003810
      CALL PLOTE(M)                                                  003820
      STOP "END OF MAIN"                                             003830
      END                                                            003840
C                                                                    003850
C                                                                    003860
      SUBROUTINE YDESIRE(Y,YSET,ALPHA,ML,YD)                         003870
      DIMENSION YD(1)                                                003880
C                                                                    003890
C  EQUATIONS PROGRAMED: YD(K)=YSET-(ALPHA**K)(YSET-Y),               003900
C  WHERE ALPHA=EXP(-TDESIRE/TAU). TAU IS THE TIME CONSTANT           003910
C  OF THE FIRST ORDER DECAYING EXPONENTIAL.                          003920
C                                                                    003930
      DO 10 K=1,ML                                                   003940
```

```
        YD(K)=YSET-(ALPHA**K)*(YSET-Y)                                  003970
10      CONTINUE                                                        003980
        RETURN                                                          003990
        END                                                             004000
C                                                                       004010
C                                                                       004020
        SUBROUTINE YZI(X,N,ML,YZ,C,FL,CFL,NDR1)                         004030
        DIMENSION X(1),YZ(1),FL(NDR1,1),CFL(1),C(1)                     004040
C                                                                       004050
C  EQUATION PROGRAMED: YZ(K)=C*(FL**K)*X                                004060
C                                                                       004070
        DO 30 I=1,ML                                                    004080
          YZ(I)=0                                                       004090
          CALL MVMAT(C,FL,N,N,CFL,NDR1)                                 004100
          DO 10 J=1,N                                                   004110
            YZ(I)=YZ(I)+CFL(J)*X(J)                                     004120
10        CONTINUE                                                      004130
          DO 20 K=1,N                                                   004140
            C(K)=CFL(K)                                                 004150
20        CONTINUE                                                      004160
30      CONTINUE                                                        004170
        RETURN                                                          004180
        END                                                             004190
C                                                                       004200
C                                                                       004210
C                                                                       004220
        SUBROUTINE CONTROL(YD,YZ,ZC,ZEQ,HEQ,QEQ,HTQHI,HTQZ,UCOMP,      004230
       +U,L,ML,NSM,NDR3)                                                004240
        DIMENSION YD(1),YZ(1),ZC(1),ZEQ(NSM,1),HEQ(NSM,1)               004250
        DIMENSION QEQ(NSM,1),HTQHI(L,1),HTQZ(1),UCOMP(1)                004260
C                                                                       004270
C  THE "NORMAL EQUATION" IS PROGRAMED AS FOLLOWS:                       004280
C  THE VECTOR OF INPUTS CALCULATED U=(HTQHI)*HTQZ.                      004290
C                                                                       004300
        DO 100 I=1,ML                                                   004310
        ZC(I)=YD(I)-YZ(I)                                               004320
100     CONTINUE                                                        004330
C  FORM HTQZ                                                            004340
        DO 120 J=1,L                                                    004350
        HTQZ(J)=0.                                                      004360
```

```
      DO 110 K=J,L                                                      004370
      HTQZ(J)=DOT3(HEQ(1,K-J+1),QEQ(1,K),ZEQ(1,K),NSM)+HTQZ(J)          004380
110   CONTINUE                                                          004390
120   CONTINUE                                                          004400
C  FORM JCOMP, THE CONTROL VECTOR                                       004410
      CALL MATV(HTQHI,HTQZ,JCOMP,L,L,7,7)                               004420
C  PICK OFF THE FIRST ELEMENT AS THE CONTROL TO                         004430
C  BE APPLIED                                                          004440
      U=JCOMP(1)                                                        004450
      RETURN                                                            004460
      END                                                               004470
C                                                                       004480
C                                                                       004490
C                                                                       004500
C                                                                       004510
      SUBROUTINE DISCRET(AT,BT,AM,BM,TDESIRE,TOBS,N,NT,NDR1,            004520
     *EAINT,FLT,GLT,FST,GST,FLM,GLM,FSM,GSM)                            004530
      DIMENSION AT(NDR1,1),BT(1),AM(NDR1,1),BM(1)                       004540
      DIMENSION EAINT(NDR1,1),FLT(NDR1,1),GLT(1),FST(NDR1,1)            004550
      DIMENSION GST(1),FLM(NDR1,1),GLM(1),FSM(NDR1,1),GSM(1)            004560
      COMMON/MAIN1/NDIM,NDIM1,JCM1/INOU/KIN,KOUT,KPUNCH                 004570
C                                                                       004580
C  THIS SUB DISCRETIZES THE A AND B MATRICES USING                      004590
C  ROUTINES ON A PACKAGE CALLED "CONTROL".                              004600
C  THE SAMPLE TIME OF TDESIRE IS HOW OFTEN THE CONTROL INPUT            004610
C  IS CALCULATED.  TIME TOBS IS THE DELTA T OF THE OUTPUT               004620
C  SAMPLER.                                                            004630
C                                                                       004640
      NDIM=NDR1                                                         004650
      NDIM1=NDR1+1                                                      004660
C  FOR THE "TRUTH" SYSTEM                                               004670
      CALL DSORT(N,AT,TDESIRE,FLT,EAINT,NT)                             004680
      CALL MATV(EAINT,BT,GLT,N,N,NDR1,NDR1)                             004690
      CALL DSORT(N,AT,TOBS,FST,EAINT,NT)                                004700
      CALL MATV(EAINT,BT,GST,N,N,NDR1,NDR1)                             004710
C  FOR THE "ACTUAL" SYSTEM                                              004720
      CALL DSORT(N,AM,TDESIRE,FLM,EAINT,NT)                             004730
      CALL MATV(EAINT,BM,GLM,N,N,NDR1,NDR1)                             004740
      CALL DSORT(N,AM,TOBS,FSM,EAINT,NT)                                004750
      CALL MATV(EAINT,BM,GSM,N,N,NDR1,NDR1)                             004760
```

```
      RETURN                                                            004770
      END                                                               004780
C                                                                       004790
C                                                                       004800
      SUBROUTINE WEIGHT(NSM,L,ML,QC,QFO)                                004810
      DIMENSION QC(1),QFO(NSM,L)                                        004820
C  THIS IS FOR MOD#2 GEO WTING                                          004830
C  WILL ONLY SET A CLEAN HALFSHALF BLOCK                                004840
C  DIVISION IF ML IS EVEN.                                              004850
      K=ML/2                                                            004860
      KK=K+1                                                            004870
      DO 10 I=1,K                                                       004880
        QC(I)=1.0                                                       004890
10    CONTINUE                                                          004900
      DO 20 J=KK,ML                                                     004910
        QC(J)=1.0E+06                                                   004920
20    CONTINUE                                                          004930
      RETURN                                                            004940
      END                                                               004950
C                                                                       004960
C                                                                       004970
      SUBROUTINE STATE(X,FS,FSX,GS,GSU,N,U,NDR)                         004980
      DIMENSION X(1),FS(NDR,1),GS(1)                                    004990
      DIMENSION GSU(1),FSX(1)                                           005000
C                                                                       005010
C  EQUATION CODED: X(K+1)=FS*X(K)+GS*U                                  005020
C                                                                       005040
      DO 10 I=1,N                                                       005050
        GSU(I)=GS(I)*U                                                  005060
10    CONTINUE                                                          005070
      CALL MATV(FS,X,FSX,N,N,NDR,NDR)                                   005080
C REPLACE X WITH THE NEW STATE                                          005090
      DO 20 J=1,N                                                       005100
        X(J)=FSX(J)+GSU(J)                                              005110
20    CONTINUE                                                          005120
      RETURN                                                            005130
      END                                                               005140
C                                                                       005150
C                                                                       005160
      SUBROUTINE OUTPUT(X,C,Y,N)                                        005170
```

```
      DIMENSION X(1),C(1)                                               005180
C                                                                       005190
C  EQUATION CODED: Y(K)=C*X(K)                                          005200
C                                                                       005210
      Y=0.0                                                             005220
      DO 10 I=1,N                                                       005230
        Y=Y+C(I)*X(I)                                                   005240
10    CONTINUE                                                          005250
      RETURN                                                            005260
      END                                                               005270
C                                                                       005280
C                                                                       005290
      SUBROUTINE PRINTM(NR,NC,DMAT,NDR)                                 005300
      DIMENSION DMAT(NDR,1)                                             005310
      DO 20 I=1,NR,1                                                    005320
        WRITE(6,10) I,(DMAT(I,J),J=1,NC,1)                              005330
10      FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))                               005340
20    CONTINUE                                                          005350
      RETURN                                                            005360
      END                                                               005370
C                                                                       005380
C                                                                       005390
      SUBROUTINE RVMAT(RV,AMAT,NR,NC,VMAT,NDR)                          005400
      DIMENSION RV(1),AMAT(NDR,1),VMAT(1)                               005410
      DO 20 I=1,NR                                                      005420
        VMAT(I)=0.0                                                     005430
        DO 10 J=1,NC                                                    005440
          VMAT(I)=VMAT(I)+RV(J)*AMAT(J,1)                               005450
10      CONTINUE                                                        005460
20    CONTINUE                                                          005470
      RETURN                                                            005480
      END                                                               005490
C                                                                       005500
C                                                                       005510
      SUBROUTINE PRINTR(NE,RV,NDR)                                      005520
      DIMENSION RV(NDR)                                                 005530
      J=1                                                               005540
      WRITE(6,10) J,(RV(I),I=1,NE)                                      005550
10    FORMAT(1H0,12,(T4,4(1PE13.4,1X)))                                 005560
      RETURN                                                            005570
```

```
      END                                                               005580
C                                                                       005590
C                                                                       005600
      SUBROUTINE MATV(AMAT,V,AMATV,NR,NC,NDR,NDC)                       005610
      DIMENSION AMAT(NDR,NDC),V(NDC),AMATV(NDR)                         005620
      DO 2) I=1,NR                                                      005630
        AMATV(I)=0.0                                                    005640
        DO 1) J=1,NC                                                    005650
          AMATV(I)=AMATV(I)+AMAT(I,J)*V(J)                              005660
1)      CONTINUE                                                        005670
2)    CONTINUE                                                          005680
      RETURN                                                            005690
      END                                                               005700
C                                                                       005710
C                                                                       005720
      SUBROUTINE HGRAPH(X,Y,N,ID,NO,NP,NS)                              005730
      DIMENSION X(1),Y(1),ID(1) $ IF(NO.EQ.2) GO TO 30                  005740
      IF (NO.LT.0) GO TO 1)                                             005750
      CALL SCALE(X,7.,N,1)       $ CALL SCALE(Y,5.,N,1)                 005760
1)    CALL PLOT(9.5,0.,-3) !   CALL PLOT(0.,11.,3)                     005770
      CALL PLOT(-1.35,1.35,3)                                           005780
      CALL PLOT(-7.15,1.35,2)  $   CALL PLOT(-7.15,9.65,2)              005790
      IF(ID(1).EQ.999) GO TO 25                                         005800
      CALL PLOT(-7.55,9.55,2)     $ CALL PLOT(-7.55,7.55,2)             005810
      DO 2) I=1,7,2                                                     005820
2)    CALL SYMBOL(1*.1-6.9,7.55 ,.07,ID(I),90.,20)                      005830
      CALL PLOT(-7.55,7.55,2)     $ CALL PLOT(-6.05,7.55,2)             005840
      CALL PLOT(-6.05,9.55,2)     $ CALL PLOT(-7.55,9.55,2)             005850
      CALL PLOT(-7.15,9.65,3)                                           005860
2)    CALL PLOT(-1.35,9.65,2)    $ CALL PLOT(-1.35,1.35,2)              005870
      CALL SYMBOL(-6.65,1.15,.1,ID(13),0.,50)                           005880
      CALL AXIS(-1.65,2.1,ID(9),-20,7.,90.,X(N+1),X(N+2))               005890
      CALL AXIS(-1.65,2.1,ID(11),20,5.,180.,Y(N+1),Y(N+2))              005900
3)    Y(I+2)=-Y(N+2)                                                    005910
      X(I+1)=X(N+2)-2.1*X(N+2) $ Y(N+1)=Y(N+1)+1.65*Y(N+2)              005920
      CALL LINE(Y,X,N,1,NP,NS)                                          005930
      X(I+1)=X(N+1)+2.1*X(N+2) $ Y(N+1)=Y(N+1)-1.65*Y(N+2)              005940
      Y(I+2)=-Y(N+2)                                                    005950
      RETURN                     $ END                                  005960
      SUBROUTINE VGRAPH(X,Y,N,ID,NO,NP,NS)                              005970
```

```
      DIMENSION X(1),Y(1),ID(1) $ IF(NO.EQ.2) GO TO 30                  000080
      IF (NJ.LT.0) GO TO 10                                             000090
      CALL SCALE(Y,4.9,N,1)       $ CALL SCALE(Y,7.0,N,1)               000100
10    CALL PLOT(8.5,0.,-3) $   CALL PLOT(0.,11.,3)                      000110
      CALL PLOT(-1.35,1.35,3)                                           000120
      CALL PLOT(-7.15,1.35,2)  $  CALL PLOT(-7.15,9.65,2)               000130
      CALL PLOT(-1.35,9.65,2) $ IF(ID(1).EQ.0H0) GO TO 25               000140
      CALL PLOT(-1.45,9.55,3) $ CALL PLOT(-3.45,9.55,2)                 000150
      DO 2) I= 1,7,2                                                    000160
20    CALL SYMBOL(-3.15,9.4-I*.10,.07,ID(I),0.,20)                      000170
      CALL PLOT(-7.45,9.55,3) $ CALL PLOT(-3.45,9.55,2)                 000180
      CALL PLOT(-1.45,9.55,2) $ CALL PLOT(-1.45,9.55,2)                 000190
      CALL PLOT(-1.35,9.65,3)                                           000200
25    CALL PLOT(-1.35,1.35,3)                                           000210
      CALL SYMBOL(-3.65,1.15,.1,ID(13),0.,20)                           000220
      CALL AXIS(-6.4,1.35,ID(9),-20,4.9,0.,X(N+1),X(N+2))               000230
      CALL AXIS(-6.4,1.35,ID(11),20,7.0,90.,Y(N+1),Y(N+2))              000240
30    X(N+1)=X(N+1)+6.4*X(N+2)  $ Y(N+1)=Y(N+1)-1.65*Y(N+2)             000250
      CALL LINE(X,Y,N,1,NP,NS)                                          000260
      X(N+1)=X(N+1)-6.4*X(N+2)  $ Y(N+1)=Y(N+1)+1.65*Y(N+2)             000270
      RETURN      $ END                                                 000280
C                                                                       000290
      FUNCTION DOT3(X,Q,Y,NSM)                                          000300
      DIMENSION X(1),Q(1),Y(1)                                          000310
      DOT3=0.                                                           000320
      DO 1 I=1,NSM                                                      000330
      DOT3=X(I)*Q(I)*Y(I)+DOT3                                          000340
1     CONTINUE                                                          000350
      RETURN                                                            000360
      END                                                               000370
C                                                                       000380
C                                                                       000390
      SUBROUTINE HBAR(C,FL,CFL,CFLK,GL,HFC,HEQ,QEQ,HTQH,HTQHI,          000400
     +N,L,ML,NSM,NDR1)                                                  000410
      DIMENSION C(1),FL(NDR1,1),GL(1),CFL(1),CFLK(1)                    000420
      DIMENSION HFC(1),HEQ(NSM,1),QEQ(NSM,1),HTQH(L,1),HTQHI(L,1)       000430
      COMMON/MAINS/NDIM,NDIM1,COM1/INOU/KIN,KOUT,KPUNCH                 000440
C                                                                       000450
C                                                                       000460
C                                                                       000470
```

```
C  NOW GET THE PIECES OF THE "NORMAL EQN", WHICH                          006390
C  IS USED TO CALCULATE THE CONTROL USED TO DRIVE                         006400
C  THE OUTPUT ALONG THE DESIRED TRAJECTORY.                               006410
C                                                                         006420
C                                                                         006430
C  THIS BLOCK(THRU SN40) CREATES AN EQUIVALENT OF HBAR                    006440
C  WHEN GO TO MIMO JUST CALL RVMAT                                        006450
C  THE ELEMENTS OF THE FIRST COL OF HBAR,HFC, IS FORMED USING THE         006460
C  RELATIONSHIP HFC(I)=SUM OVER I OF(C*F** (I-1)*G).                      006470
C                                                                         006480
C                                                                         006490
      NDIM=L                                                              006500
      NDIM1=L+1                                                           006510
C  CHANGE L TO NDR3 WHEN GO TO 10TH ORDER                                 006520
      HFC(1)=0                                                            006530
      DO 10 I=1,N                                                         006540
        HFC(1)=HFC(1)+C(I)*GL(I)                                          006550
10    CONTINUE                                                            006560
C  FORM CFL                                                               006570
      CALL RVMAT(C,FL,N,N,CFL,NDR1)                                       006580
      DO 40 J=2,ML                                                        006590
        CFLGL=0.0                                                         006600
        DO 20 JJ=1,N                                                      006610
          CFLGL=CFLGL+CFL(JJ)*GL(JJ)                                      006620
20      CONTINUE                                                          006630
        HFC(J)=HFC(J-1)+CFLGL                                             006640
        CALL RVMAT(CFL,FL,N,N,CFLK,NDR1)                                  006650
C  EQUATE THE TWO MATRICES SO CAN GO THRU INNER                           006660
C  LOOP AGAIN                                                             006670
        DO 30 II=1,N                                                      006680
          CFL(II)=CFLK(II)                                                006690
30      CONTINUE                                                          006700
40    CONTINUE                                                            006710
C                                                                         006720
C                                                                         006730
C  WILL NOW FORM THE REST OF THE PIECES NEEDED FOR THE                    006740
C  'NORMAL' EQN USED IN CALCULATION OF THE CONTROL INPUT.                 006750
C  FORM ATOH                                                              006760
C  THIS IS THE REID METHOD                                                006770
      DO 70 I=1,L                                                         006780
```

```
      DO 60 J=I,L                                                       006780
      HTQH(I,J)=0.                                                      006790
      DO 50 K=J,L                                                       006800
      HTQH(I,J)=DOTJ(HEQ(1,K-I+1),QEQ(1,K),HEQ(1,K-J+1),NSM)+HTQH(I,J)  006810
50    CONTINUE                                                          006820
60    CONTINUE                                                          006830
70    CONTINUE                                                          006840
C  HAVE FORMED THE UPPER SECTION OF HEQ,OFF DIAGONAL TERMS             006850
C  ARE TRANSPOSES.  IN THE SISO CASE ELEMENTS ARE SCALARS              006860
C  SO THE OFF DIAGONAL TERMS ARE EQUAL.  THIS BLOCK                    006870
C  DEFINES THE LOWER ELEMENTS IN TERMS OF THE UPPER ONES.             006880
      DO 90 I=1,L                                                       006890
      DO 80 J=I,L                                                       006900
      IF(I.EQ.J) GO TO 80                                               006910
      HTQH(J,I)=HTQH(I,J)                                               006920
80    CONTINUE                                                          006930
90    CONTINUE                                                          006940
      CALL GMINV(L,L,HTQH,HTQHI,MR,1)                                   006950
      RETURN                                                            006960
      END                                                               006970
C                                                                       006980
C                                                                       006990
      SUBROUTINE NOISE(RMSNOIS,OUTMEAN,WN)                              007000
C                                                                       007010
C  SUB NOISE CALCULATES THE VALUES OF THE MEASUREMENT                   007020
C  NOISE USING A RANDOM NOISE GENERATOR MODELLED AS                    007030
C  GAUSSIAN.                                                           007040
C                                                                       007050
      GAUSS=0.                                                          007060
      DO 10 I=1,12                                                      007070
        GAUSS=GAUSS+RANF(DUM)                                           007080
10    CONTINUE                                                          007090
      GAUSS=GAUSS-6.+OUTMEAN                                            007100
      WN=GAUSS*RMSNOIS                                                  007110
      RETURN                                                            007120
      END                                                               007130
```

Appendix C

Graphical Results of Model Mismatch Simulation

Appendix C contains the graphical results corresponding to the cases listed in Table II and Table III in Chapter IV.

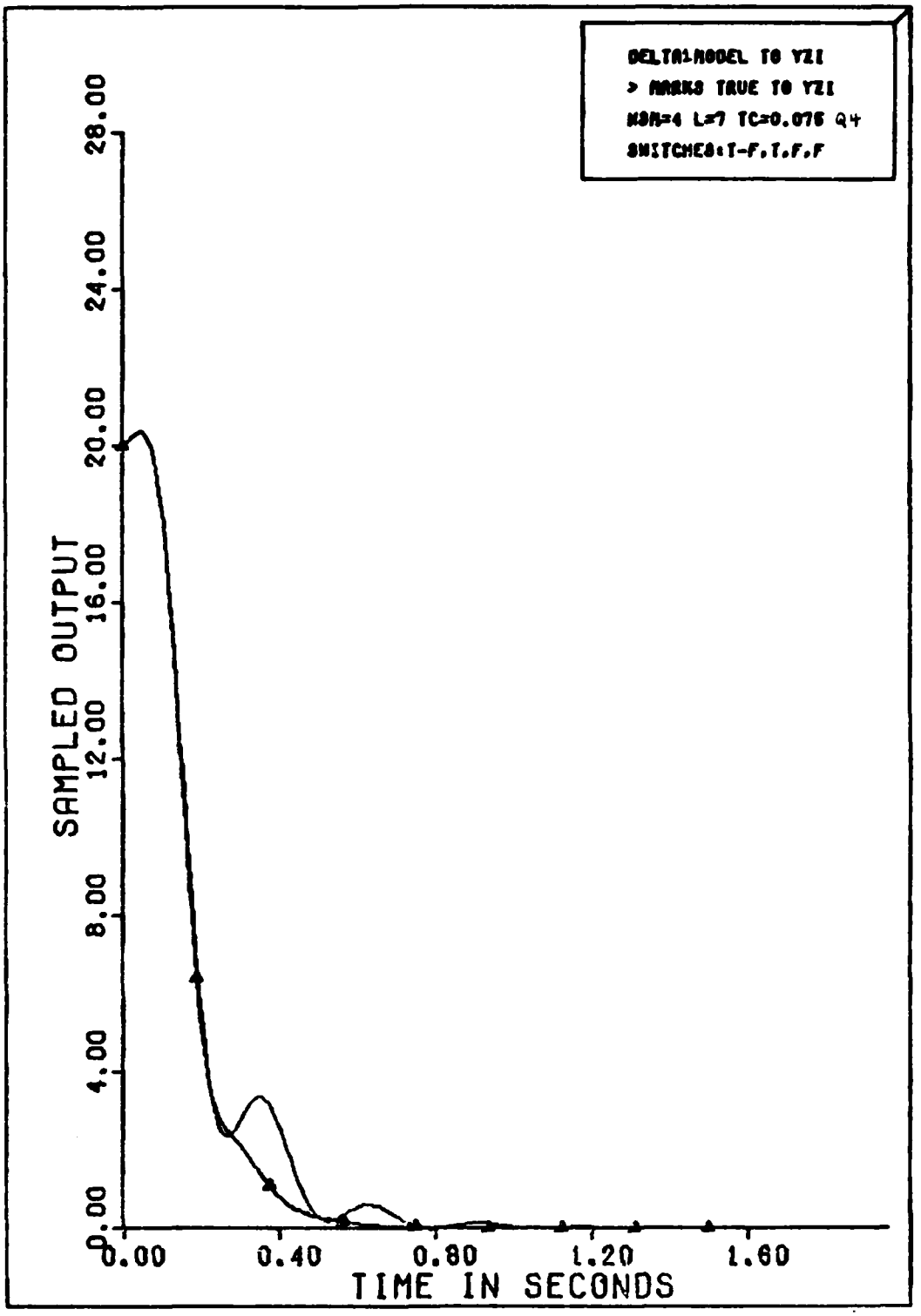


Fig C-1. System Output for Truth Model (Δ) and 10% Delta 1 Model to Zero-Input Response Calculation, TC=0.075

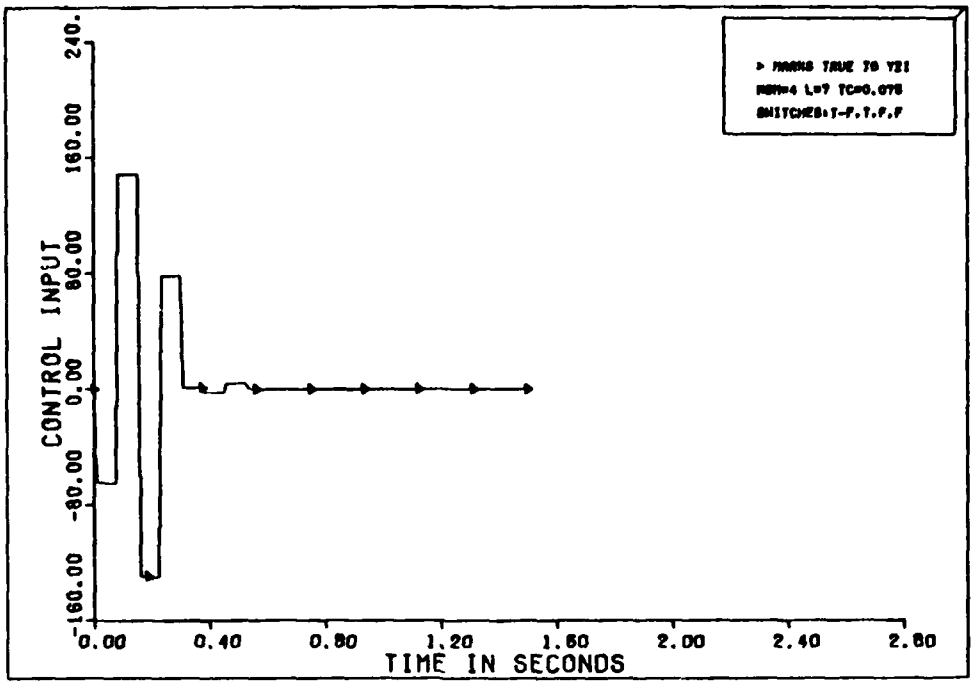


Fig C-2. Controls Applied for Truth Model to Zero-Input Response Calculation, TC=0.075

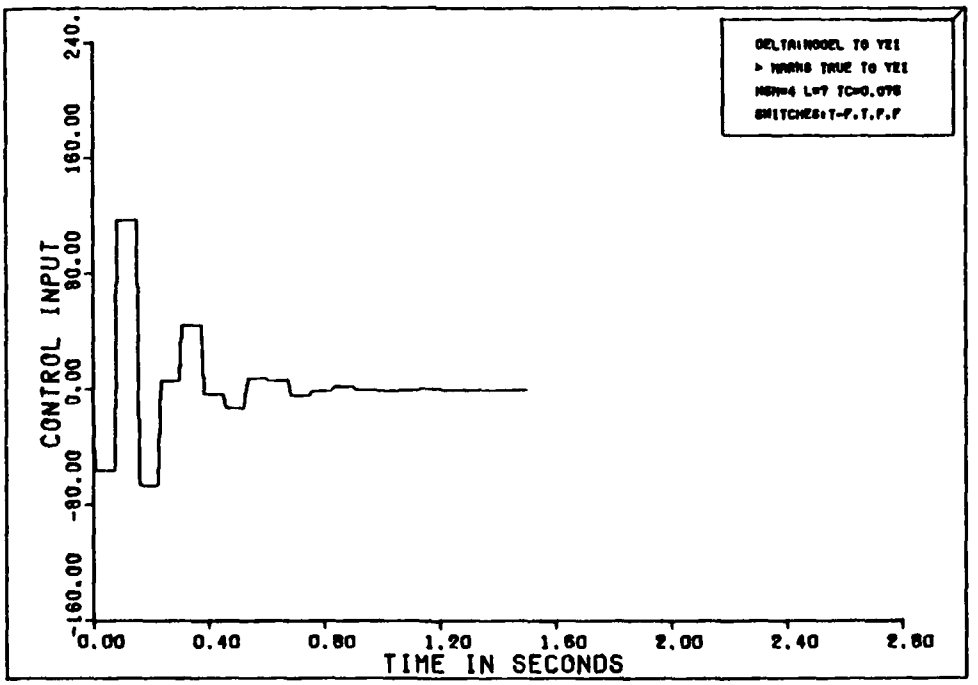


Fig C-3. Controls Applied for 10% Delta 1 Model to Zero-Input Response Calculation, TC=0.075

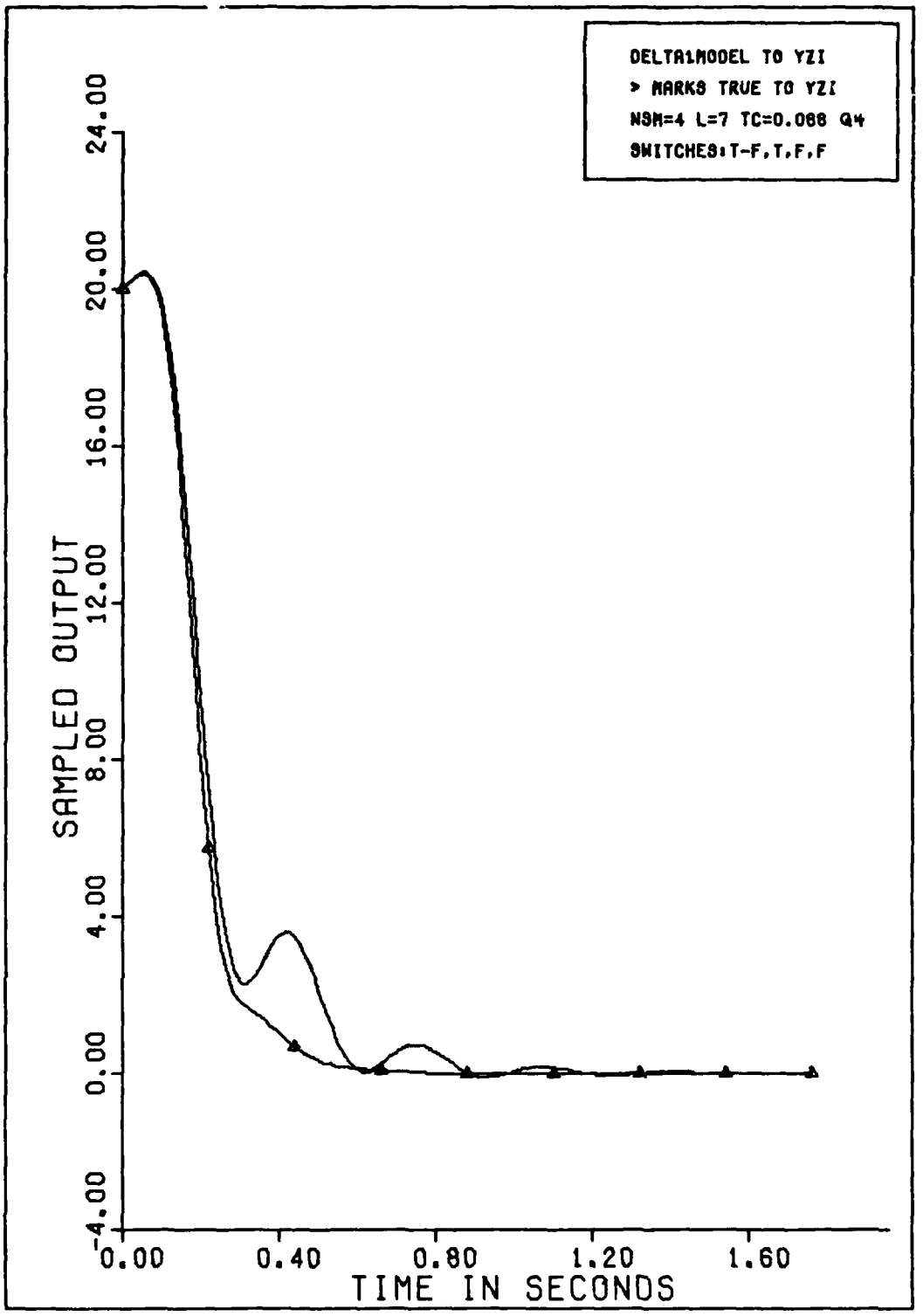


Fig C-4. System Output for Truth Model (Δ) and 10% Delta 1 Model to Zero-Input Response Calculation, TC=0.088

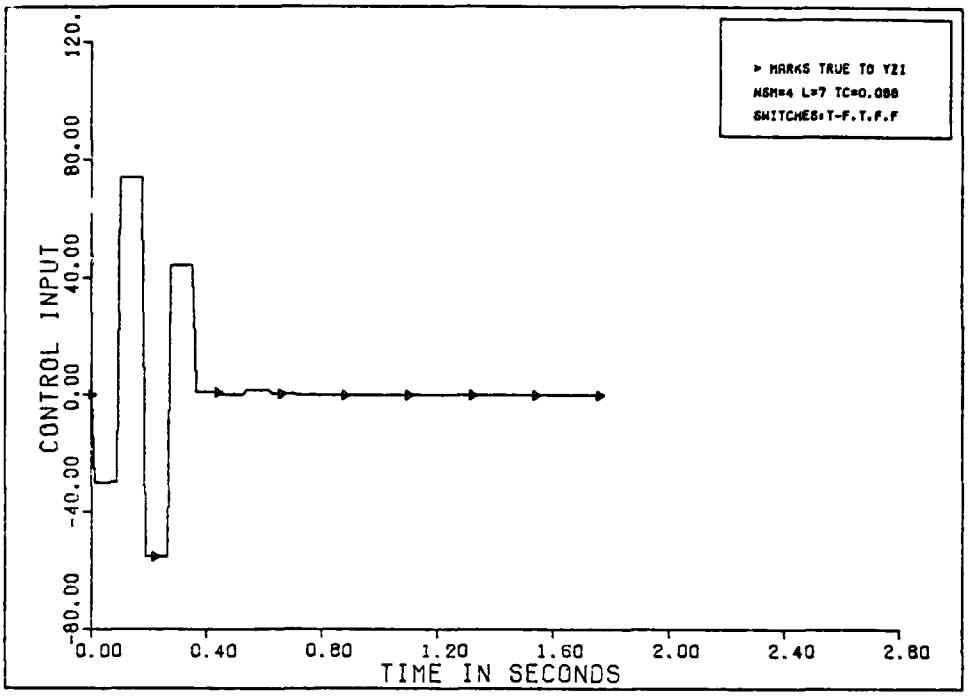


Fig C-5. Controls Applied for Truth Model to Zero-Input Response Calculation, TC=0.088

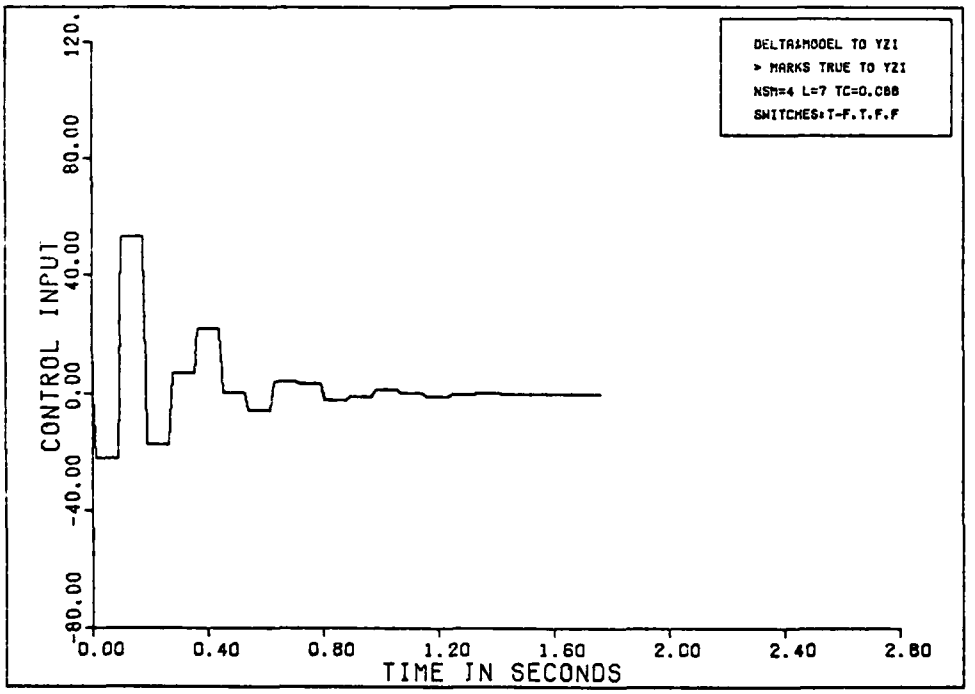


Fig C-6. Controls Applied for 10% Delta 1 Model to Zero-Input Response Calculation, TC=0.088

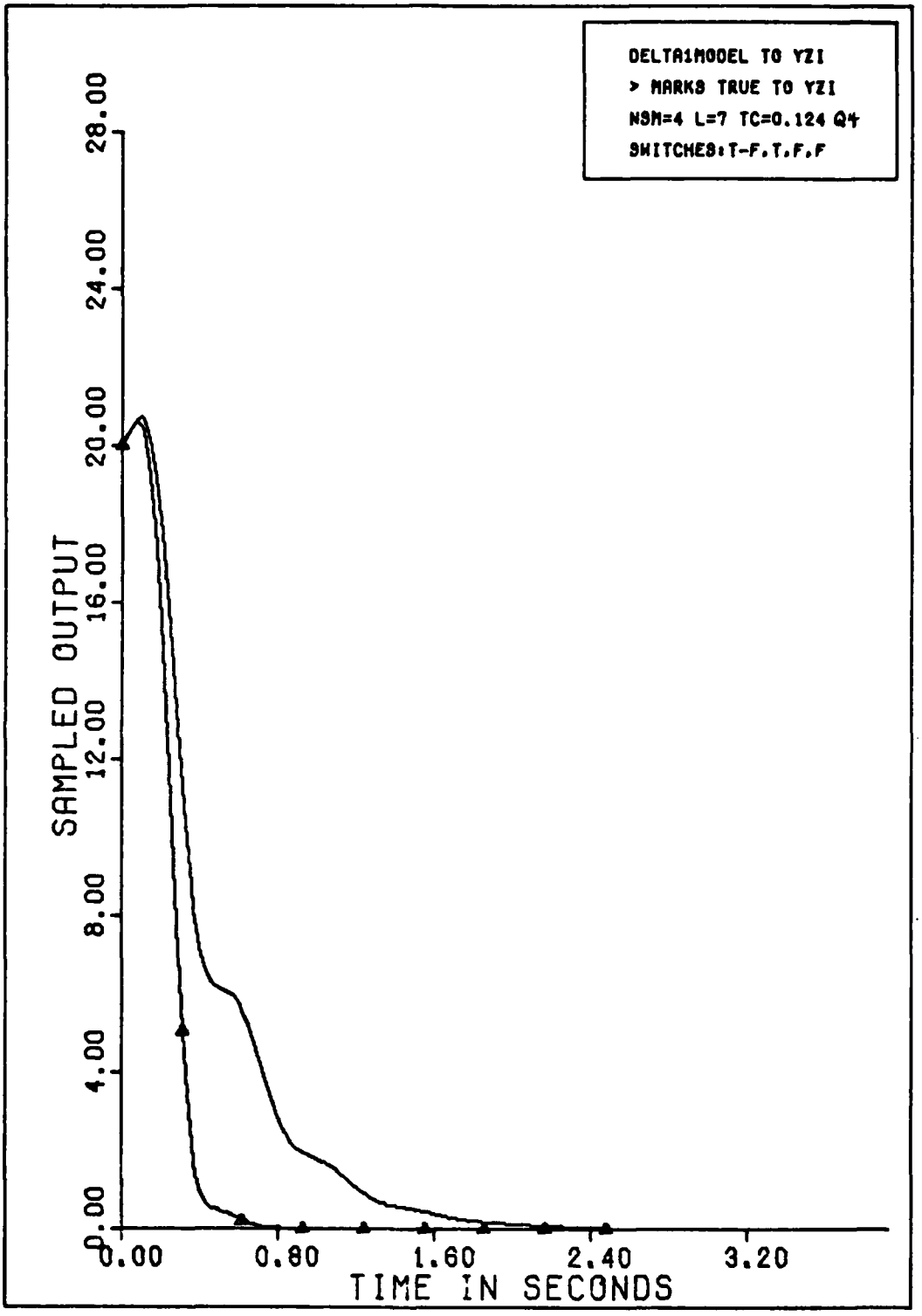


Fig C-7. System Output for Truth Model (Δ) and 10% Delta 1 Model to Zero-Input Response Calculation, TC=0.124

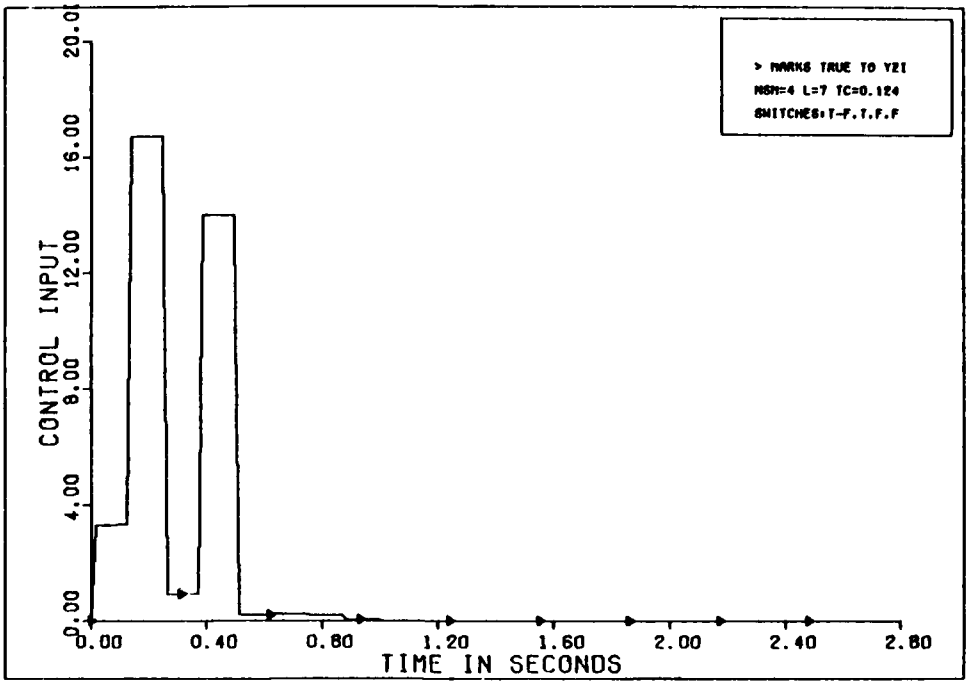


Fig C-8. Controls Applied for Truth Model to Zero-Input Response Calculation, TC=0.124

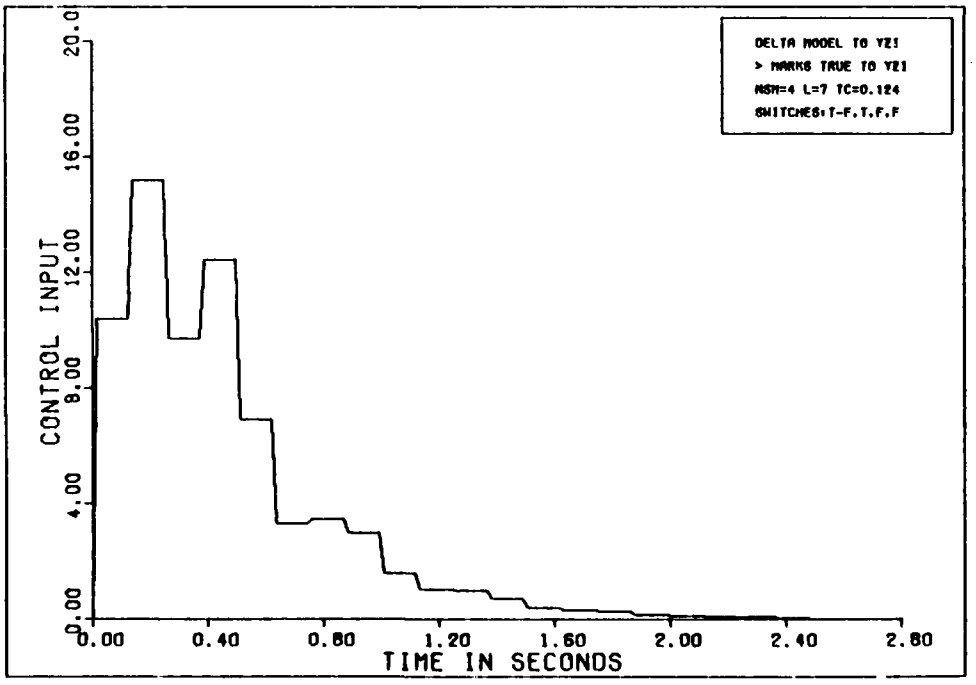


Fig C-9. Controls Applied for 10% Delta 1 Model to Zero-Input Response Calculation, TC=0.124

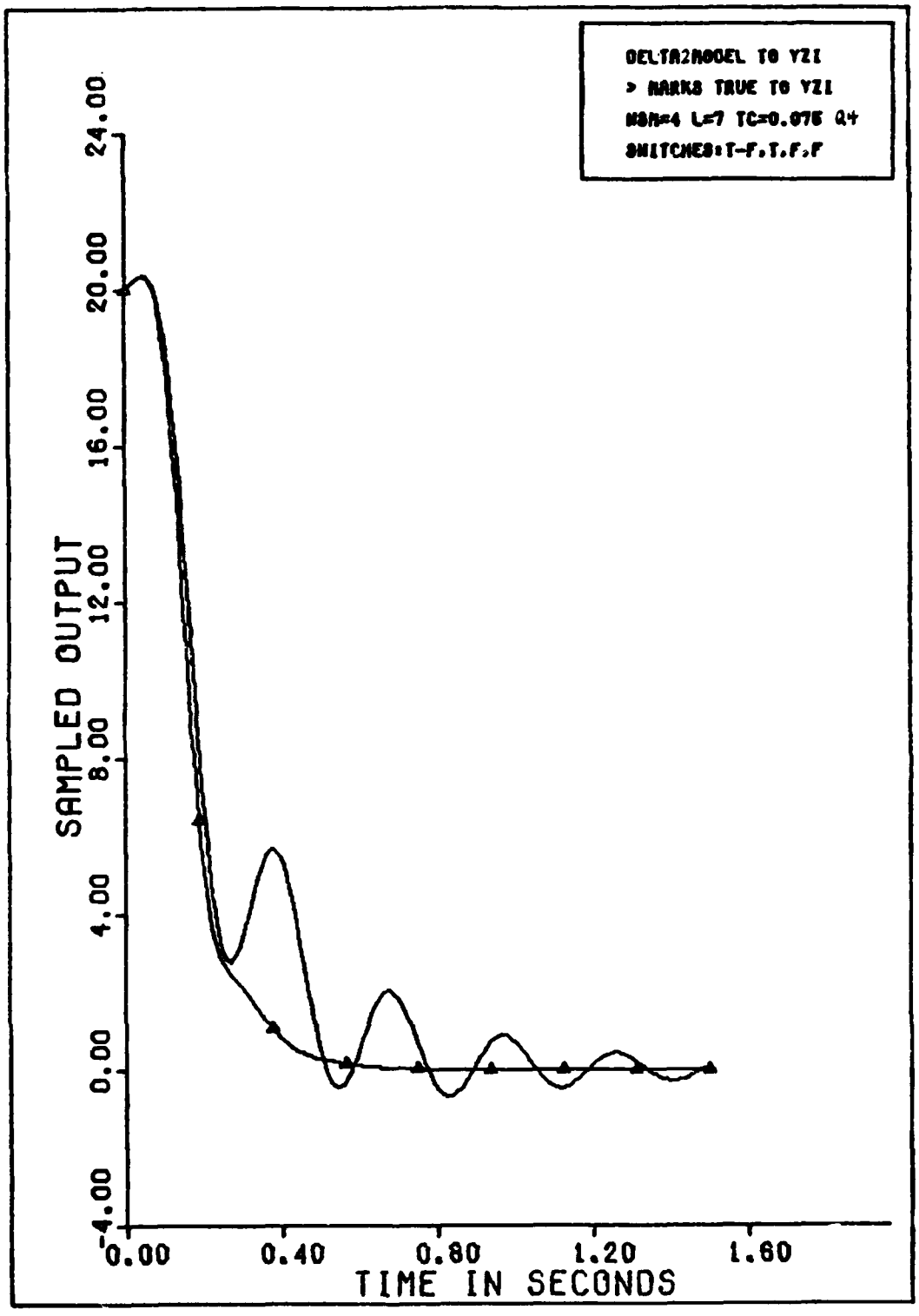


Fig C-10. System Output for Truth Model (Δ) and 20% Delta 2 Model to Zero-Input Response Calculation, TC=0.075

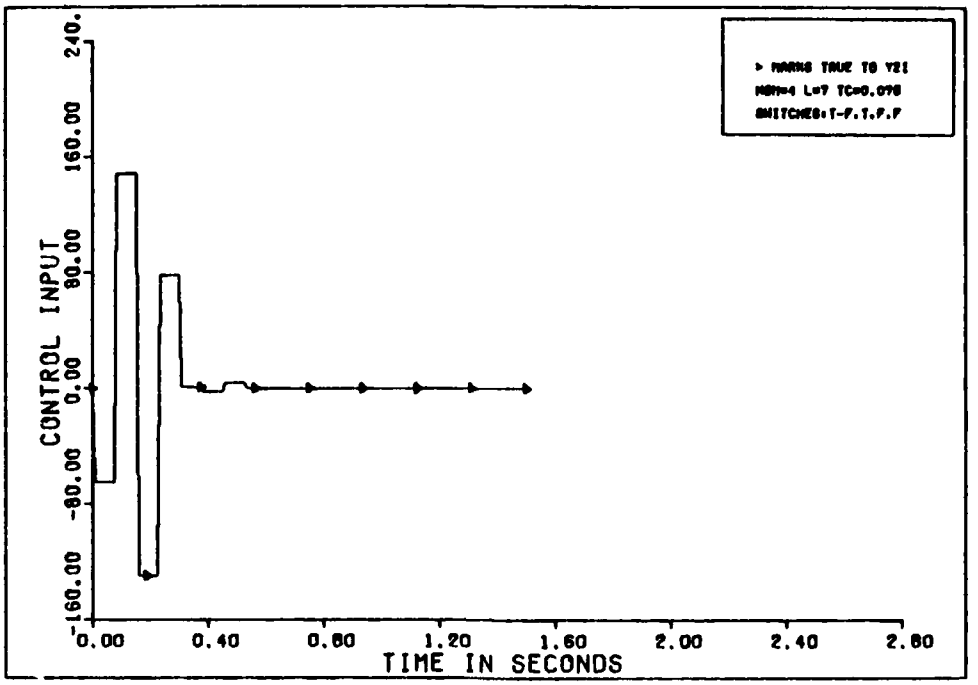


Fig C-11. Controls Applied for Truth Model to Zero-Input Response Calculation, TC=0.075

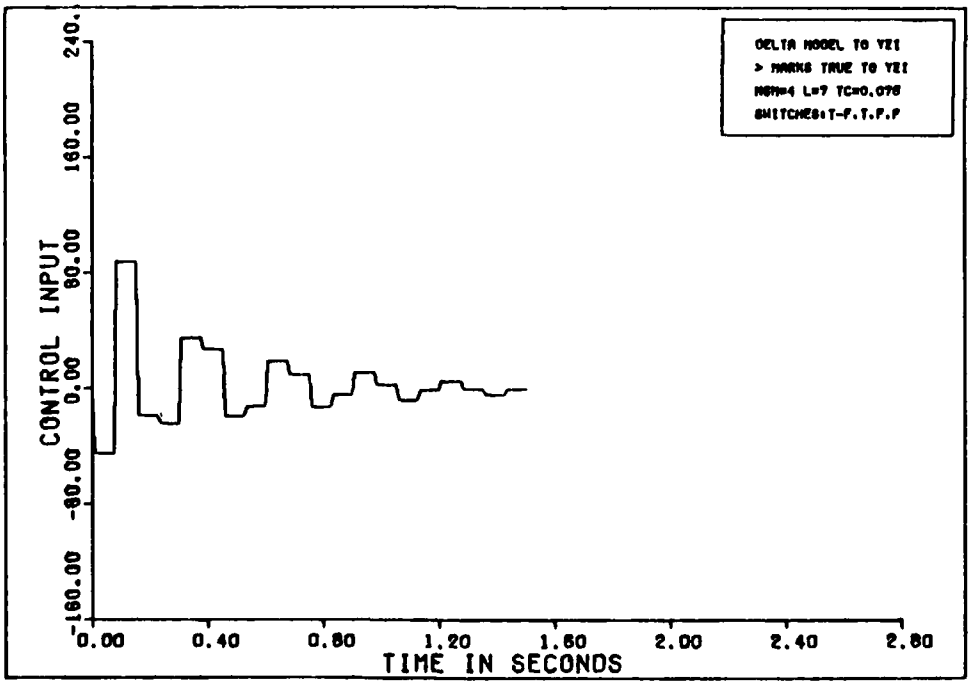


Fig C-12. Controls Applied for 20% Delta 2 Model to Zero-Input Response Calculations, TC=0.075

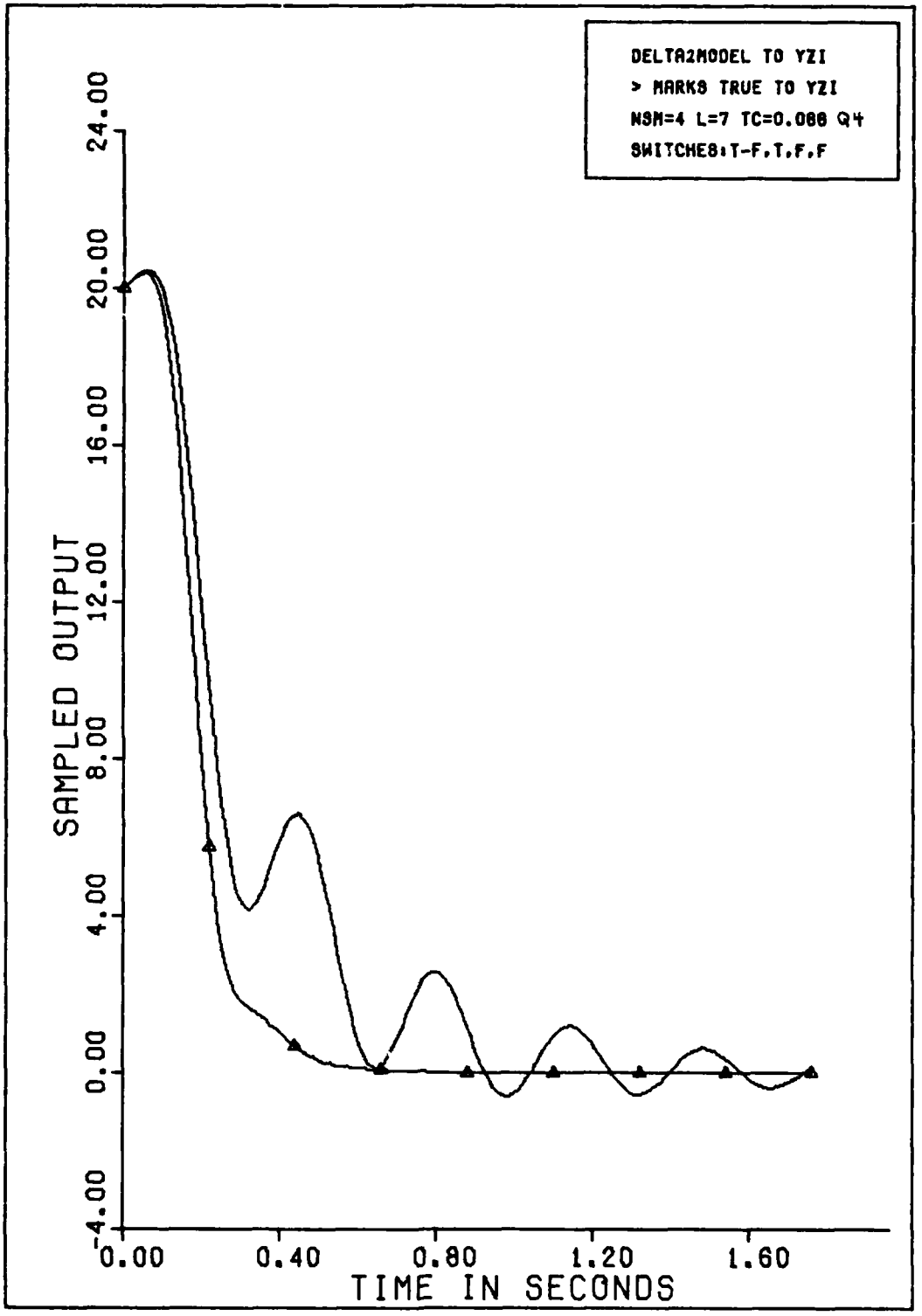


Fig C-13. System Output for Truth Model (Δ) and 20% Delta 2 Model to Zero-Input Response Calculation, TC=0.088

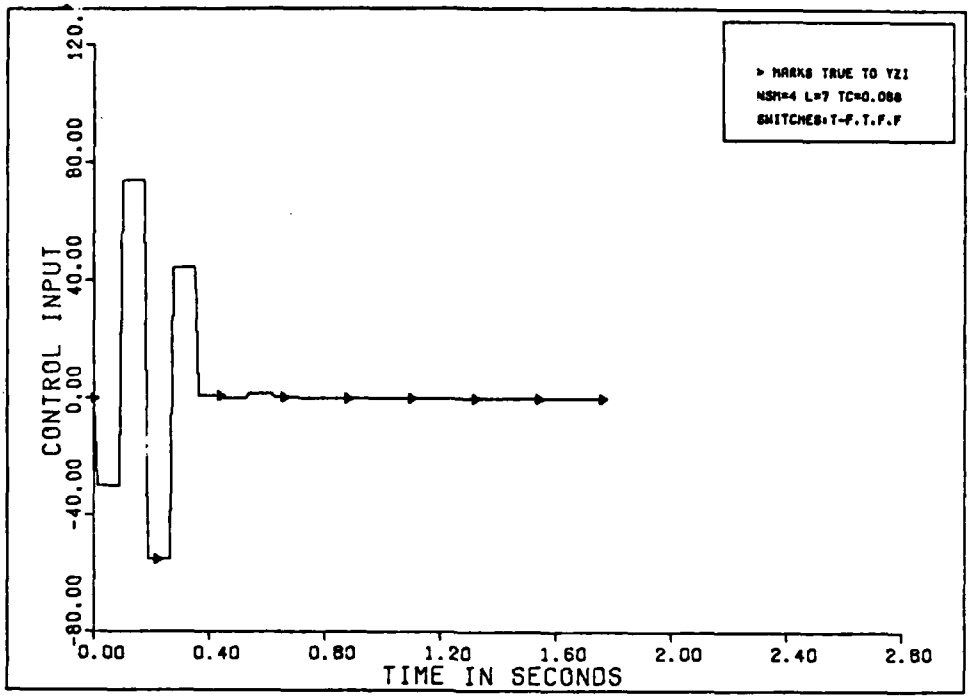


Fig C-14. Controls Applied for Truth Model to Zero-Input Response Calculation, TC=0.088

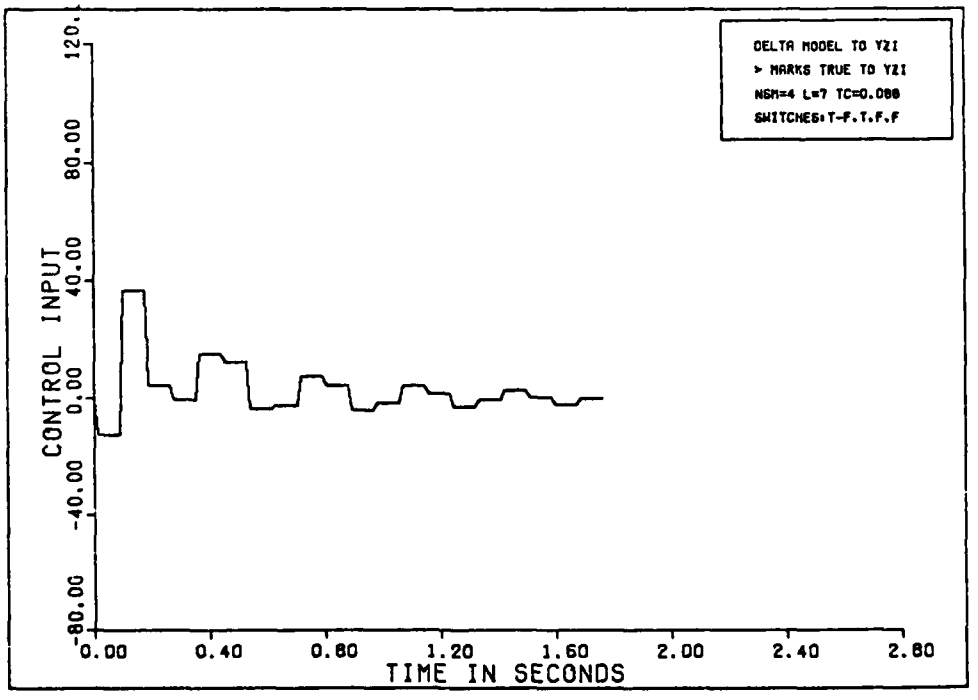


Fig C-15. Controls Applied for 20% Delta 2 Model to Zero-Input Response Calculation, TC=0.088

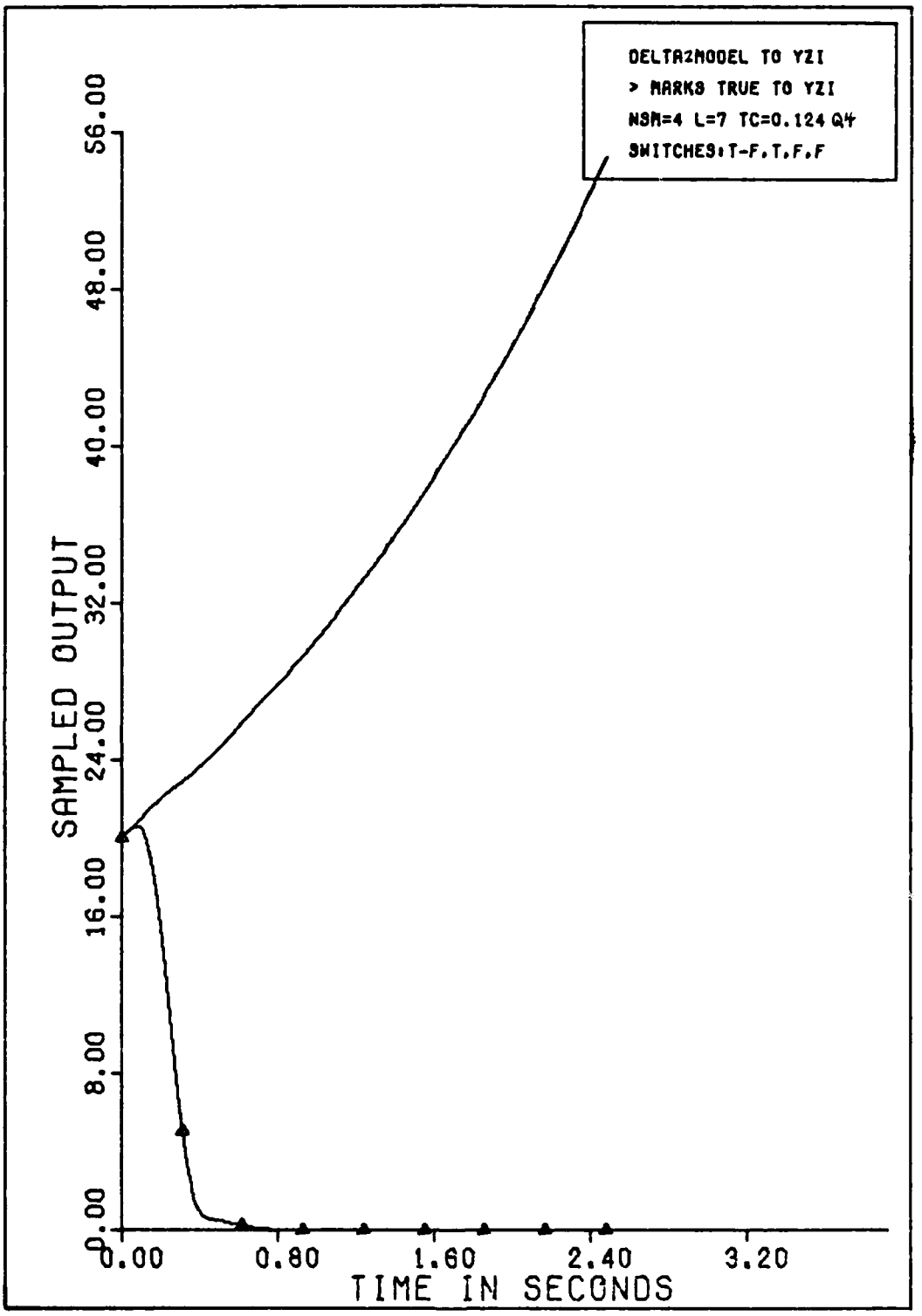


Fig C-16. System Output for Truth Model (Δ) and 20% Delta 2 Model to Zero-Input Response Calculation, TC=0.124

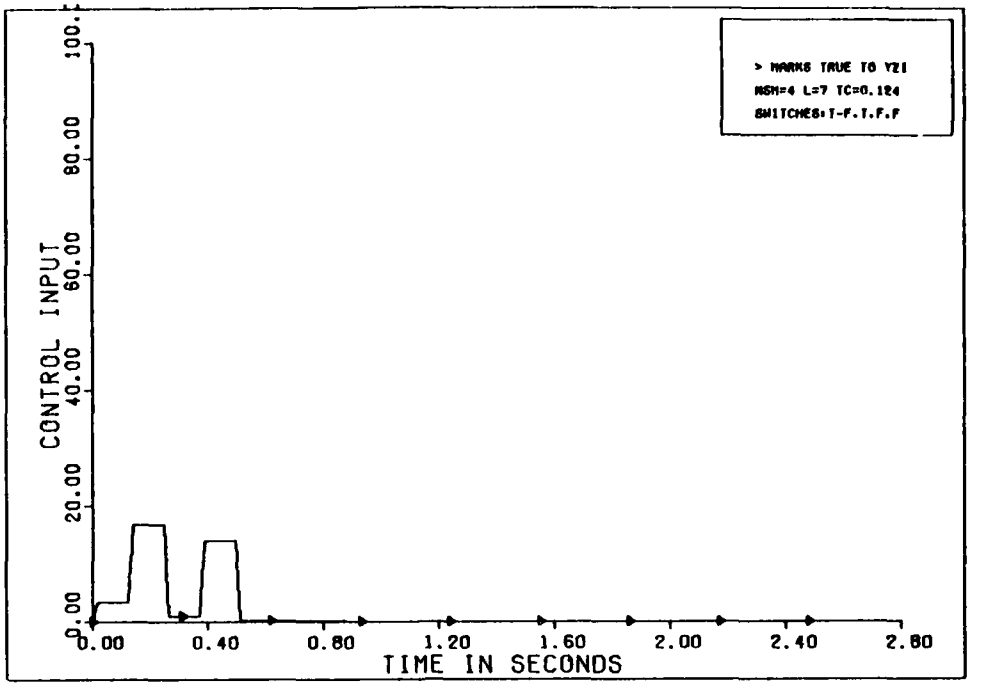


Fig C-17. Controls Applied for Truth Model to Zero-Input Response Calculation, TC=0.124

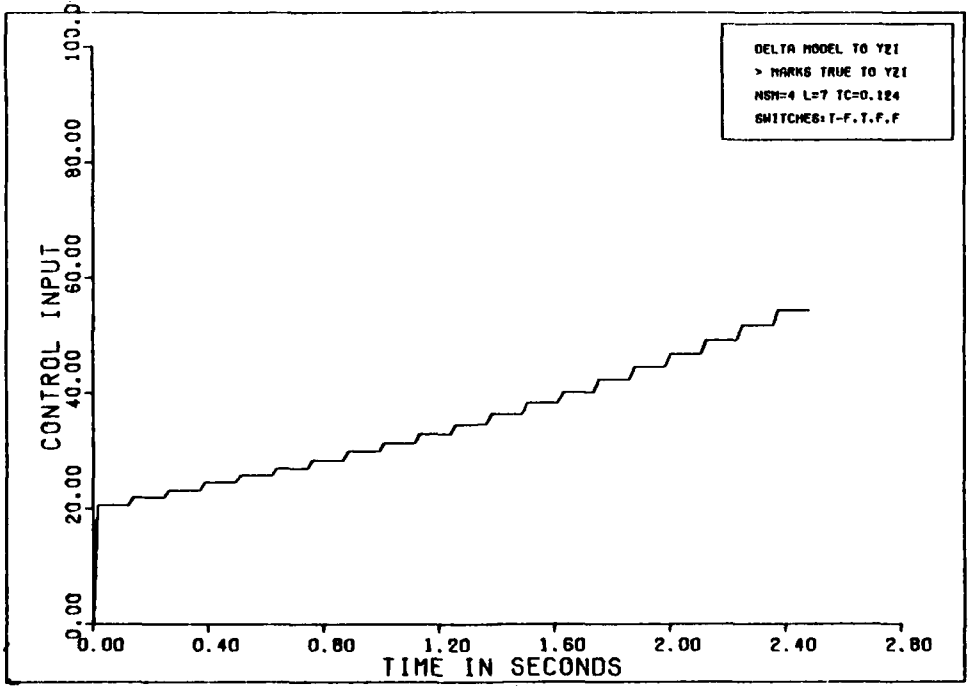


Fig C-18. Controls Applied for 20% Delta 2 Model to Zero-Input Response Calculation, TC=0.124

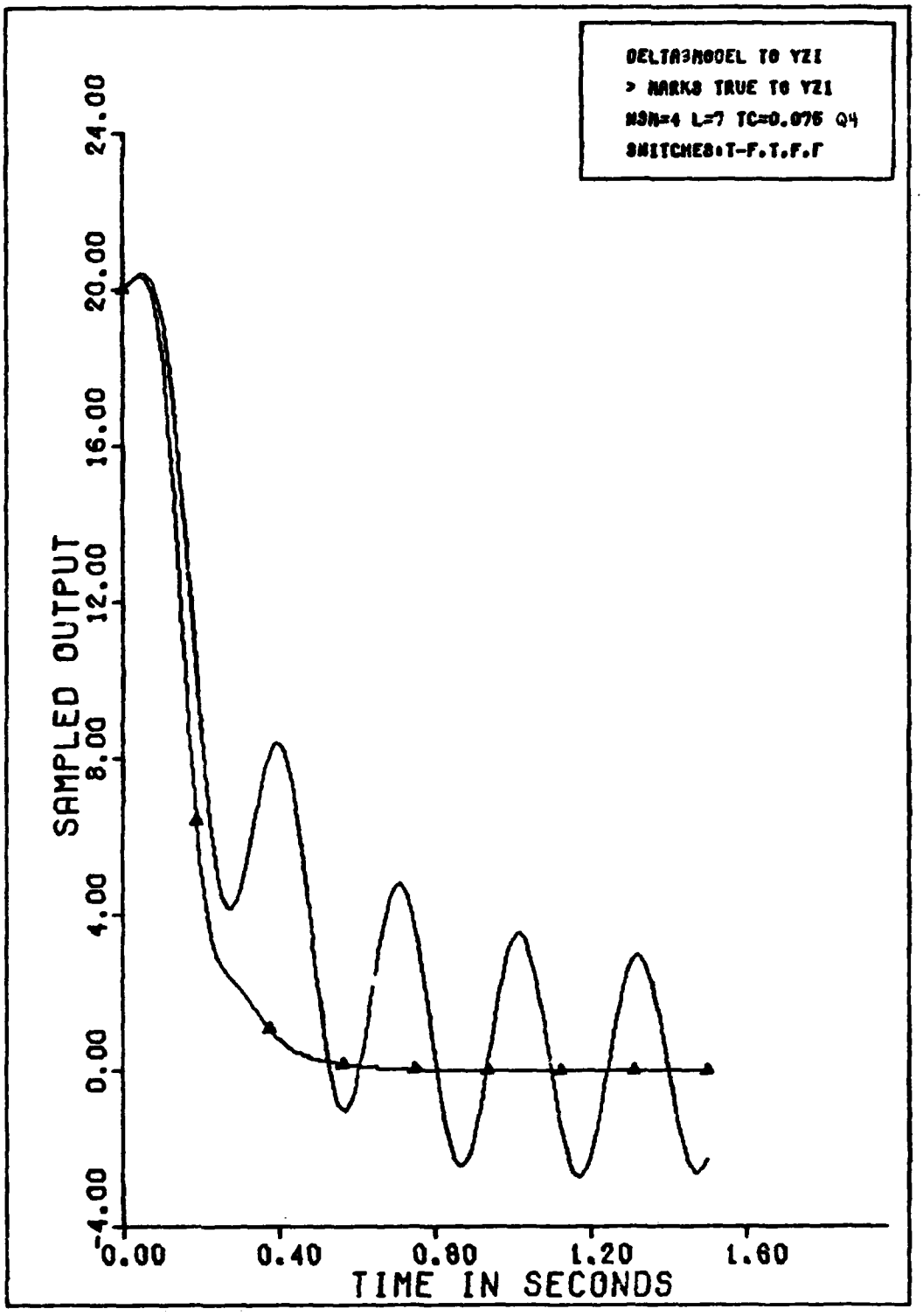


Fig C-19. System Output for Truth Model (Δ) and 30% Delta 3 Model to Zero-Input Response Calculation, TC=0.075

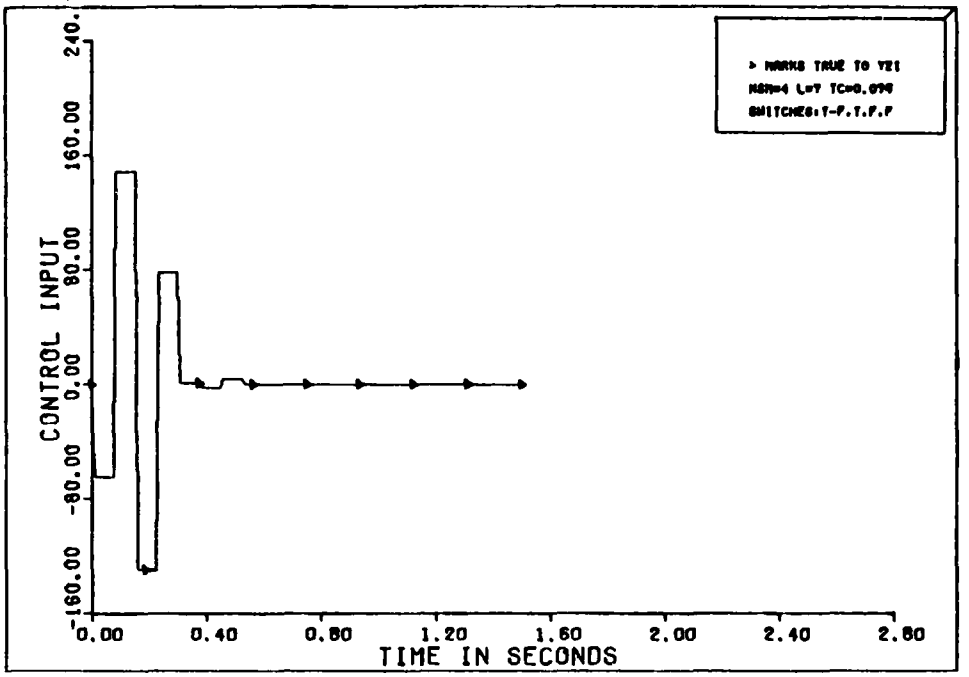


Fig C-20. Controls Applied for Truth Model to Zero-Input Response Calculation, TC=0.075

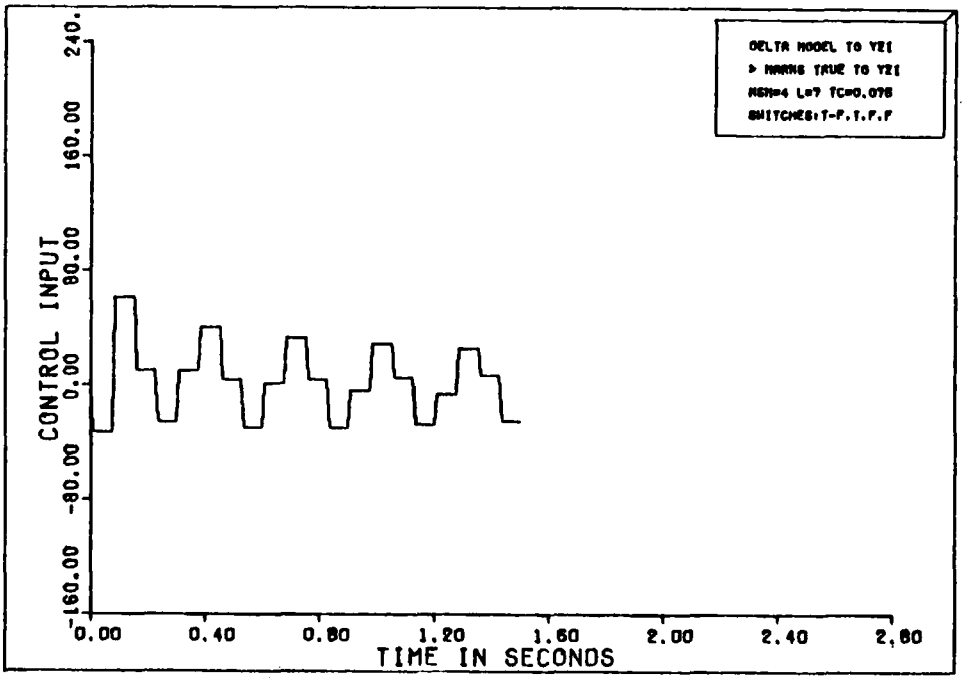


Fig C-21. Controls Applied for 30% Delta 3 Model to Zero-Input Response Calculation, TC=0.075

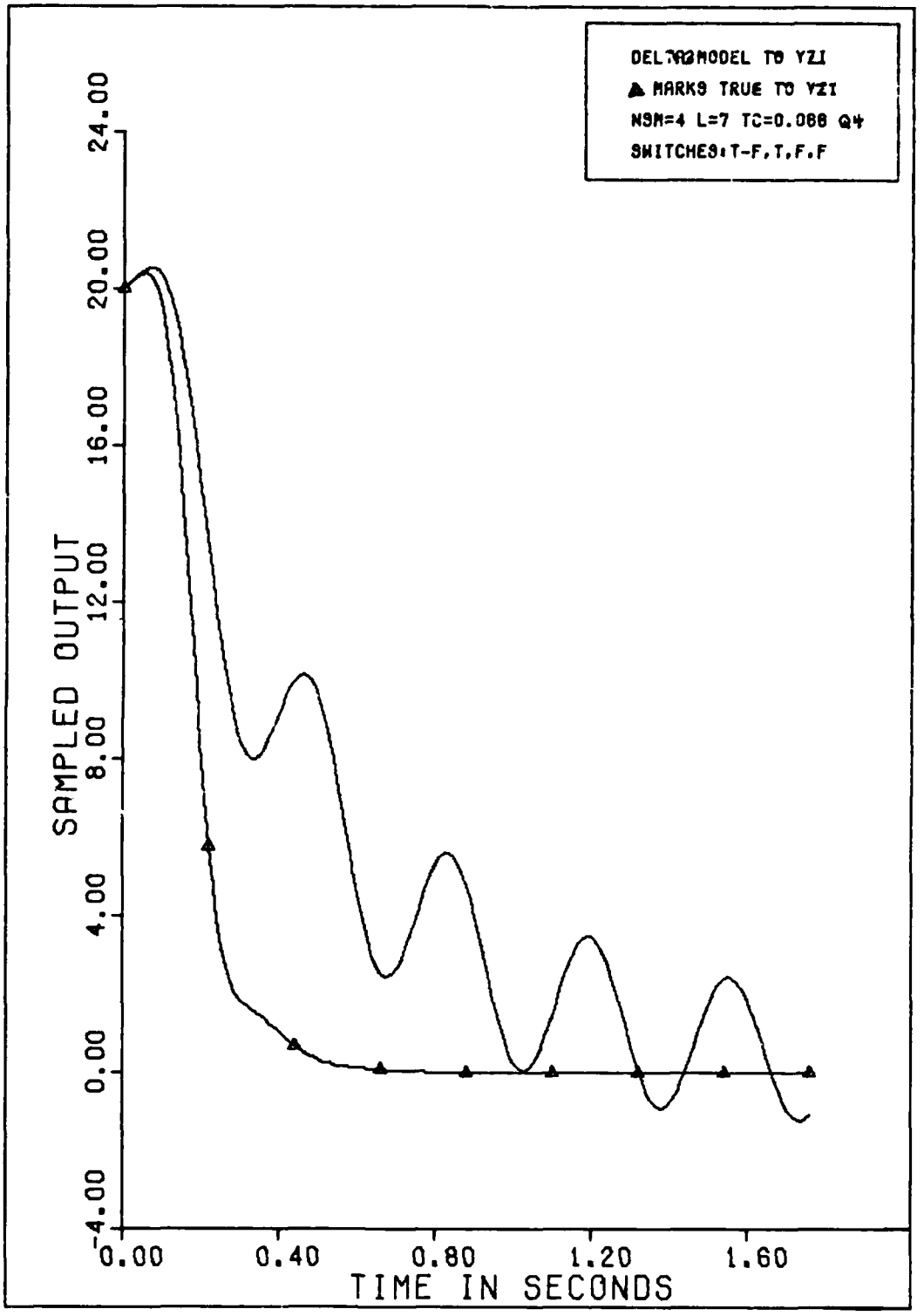


Fig C-22. System Output for Truth Model (Δ) and 30% Delta 3 Model to Zero-Input Calculation, TC=0.088

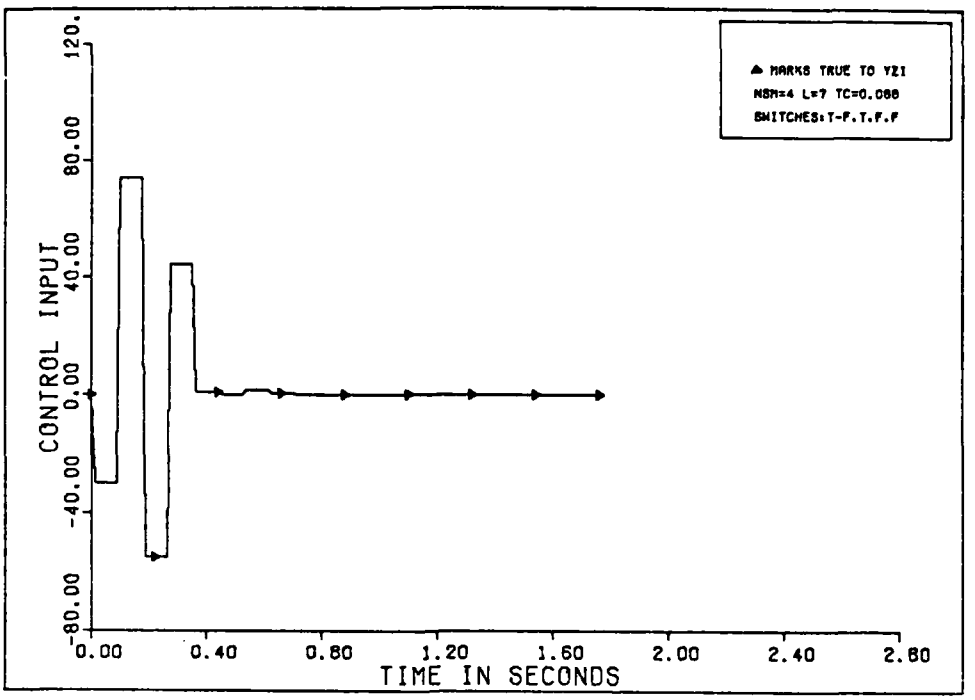


Fig C-23. Controls Applied for Truth Model to Zero-Input Response Calculation, TC=0.088

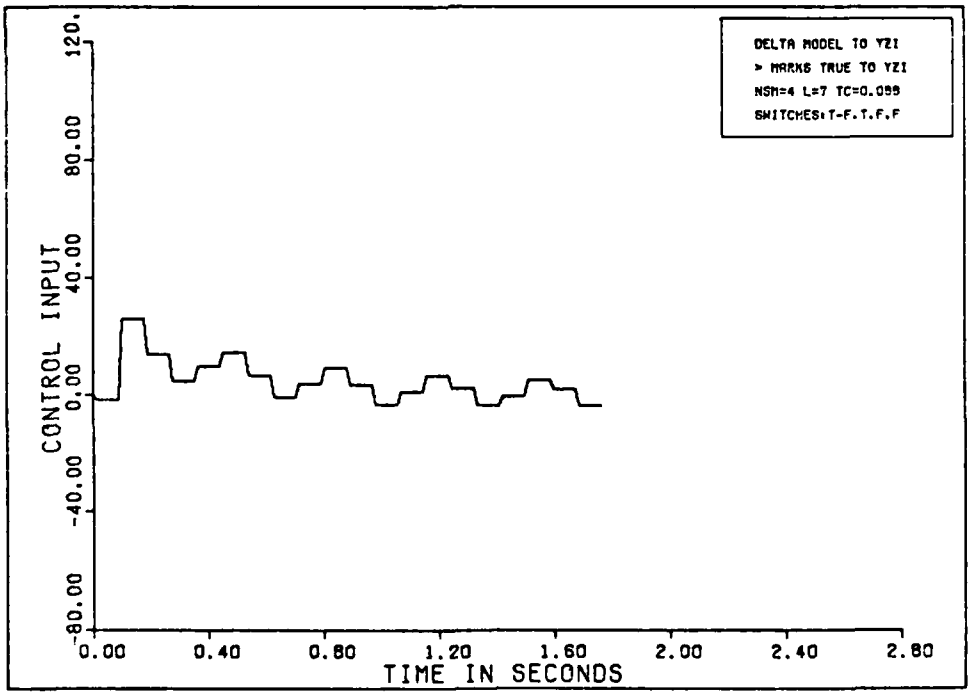


Fig C-24. Controls Applied for 30% Delta 3 Model to Zero-Input Response Calculation, TC=0.088

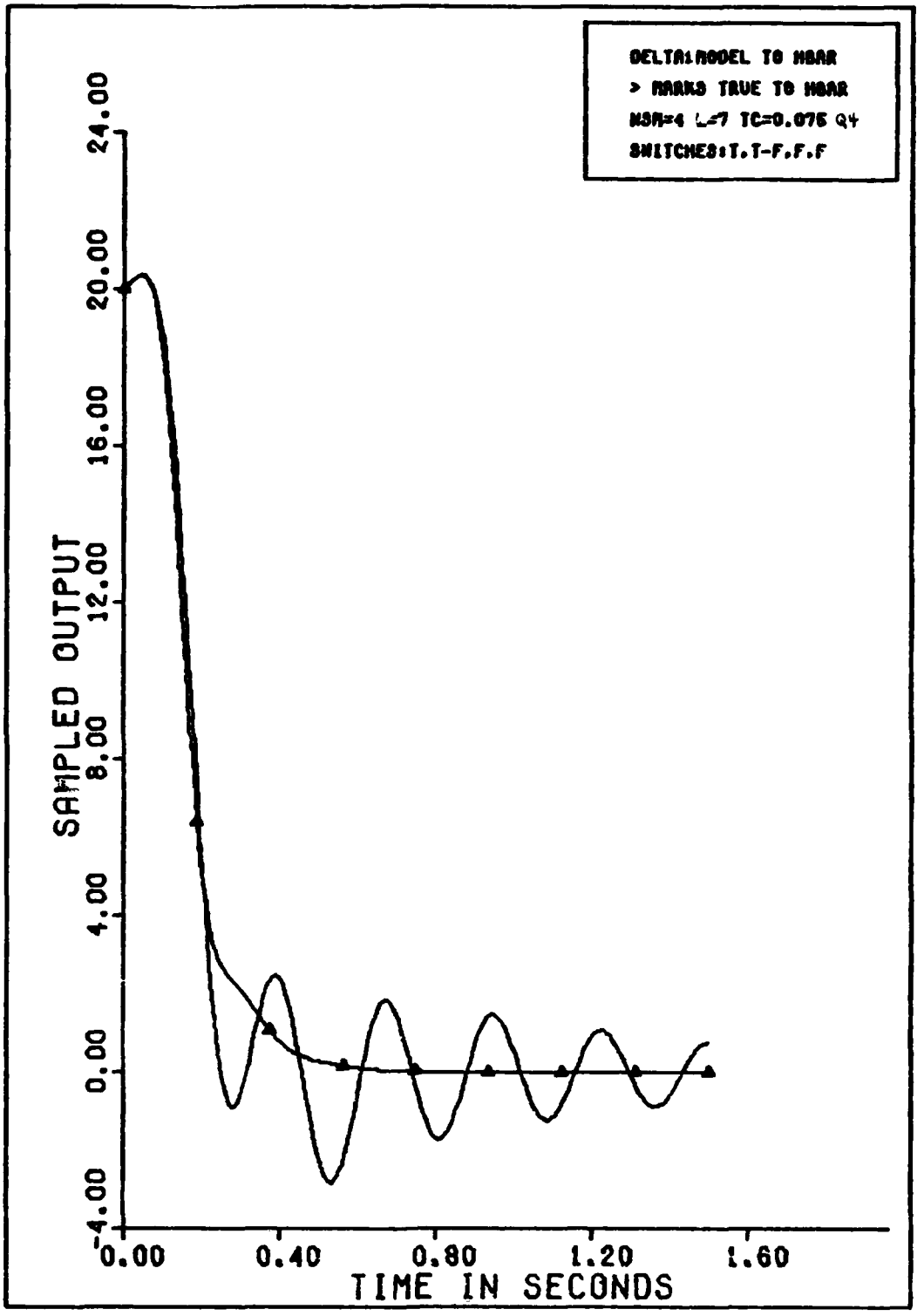


Fig C-25. System Output for Truth Model (Δ) and 10% Delta 1 Model to $\bar{H}$ Calculation, TC=0.075

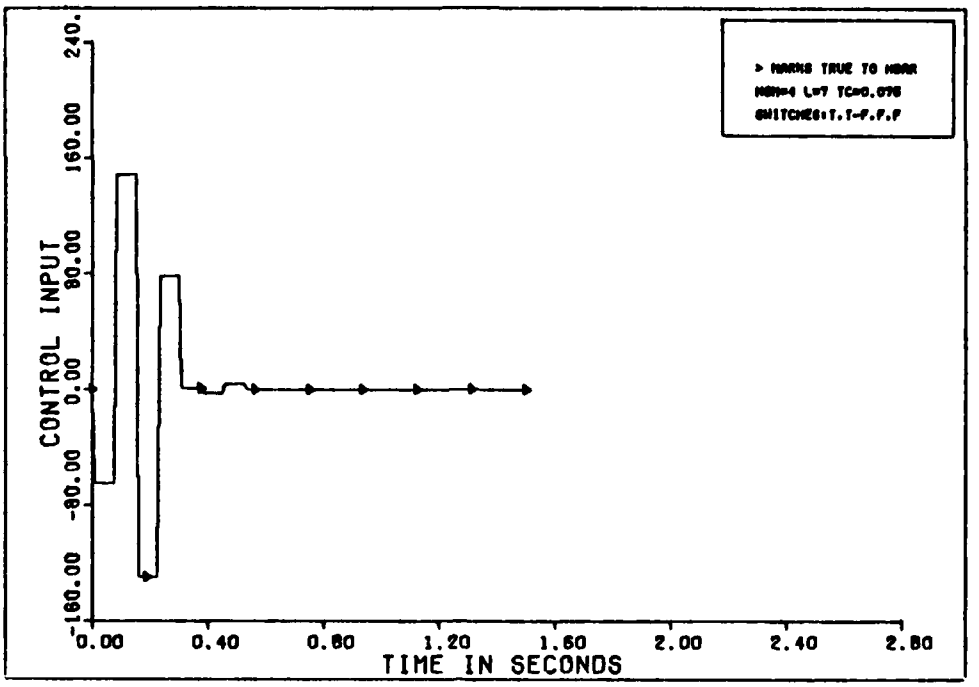


Fig C-26. Controls Applied for Truth Model to $\bar{H}$ Calculation, TC=0.075

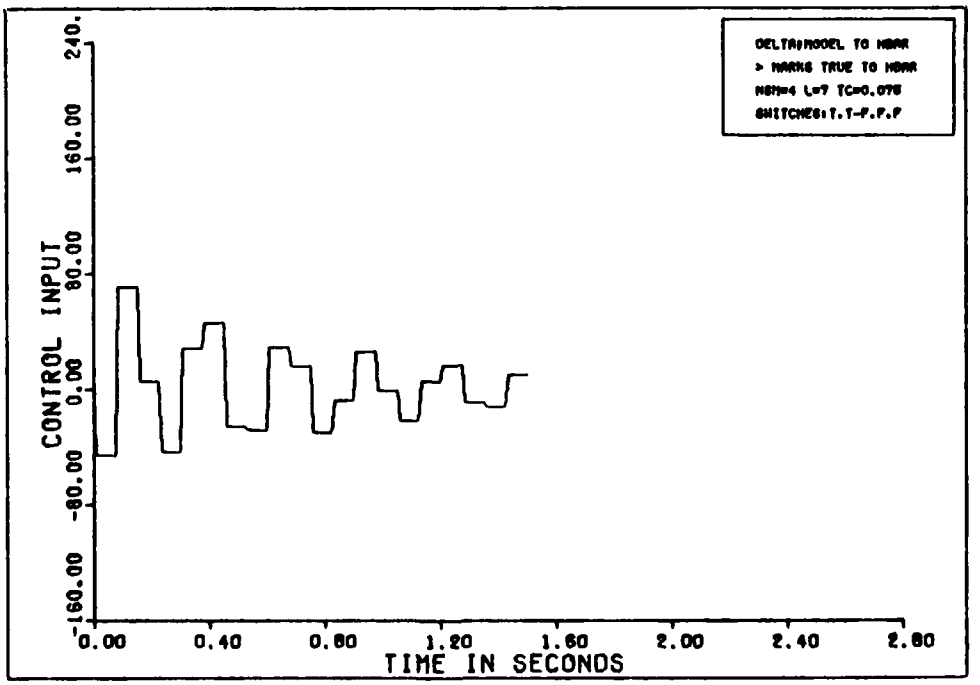


Fig C-27. Controls Applied for 10% Delta 1 Model to $\bar{H}$ Calculation, TC=0.075

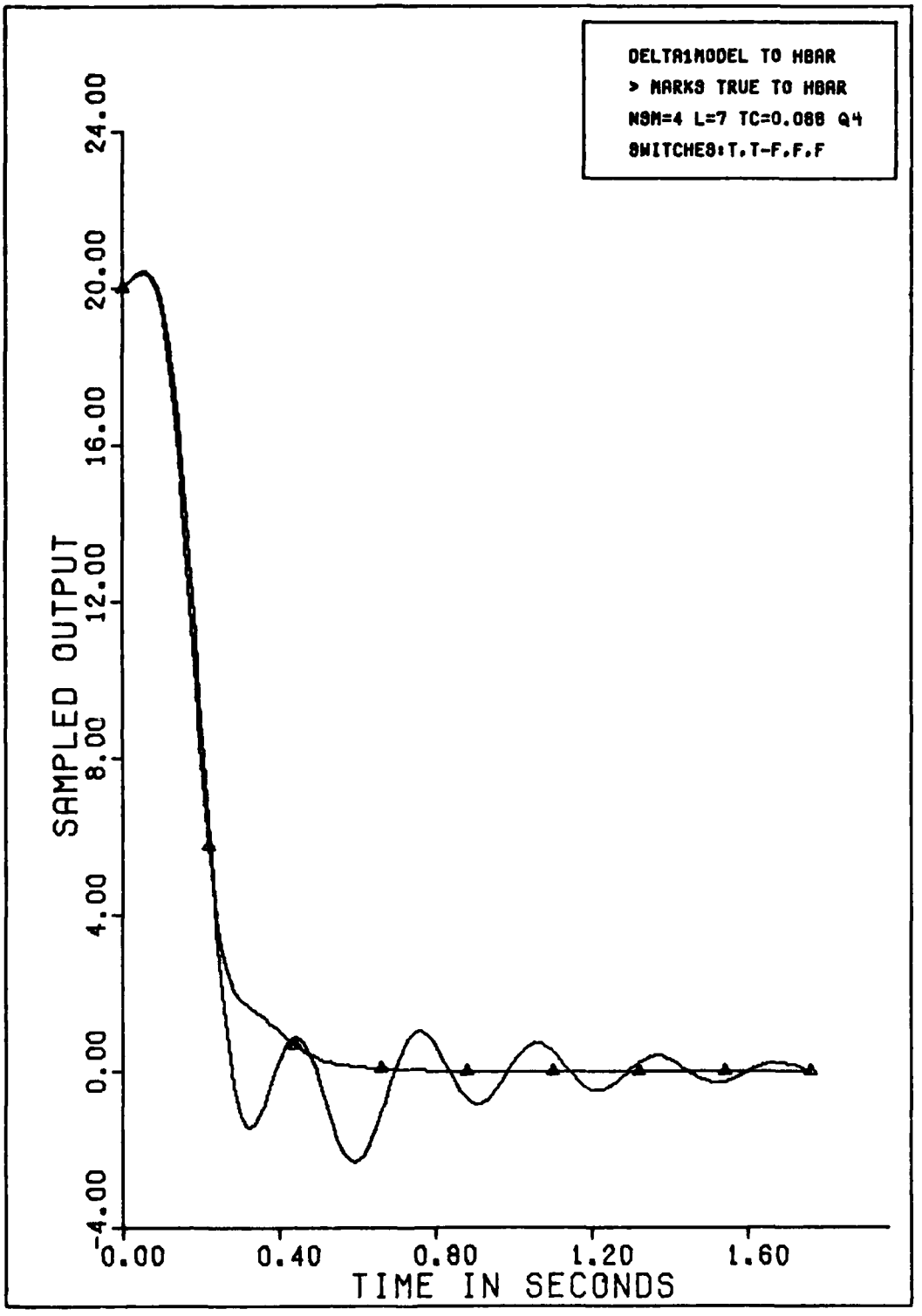


Fig C-28. System Output for Truth Model (Δ) and 10% Delta 1 Model to $\bar{H}$ Calculation, TC=0.088

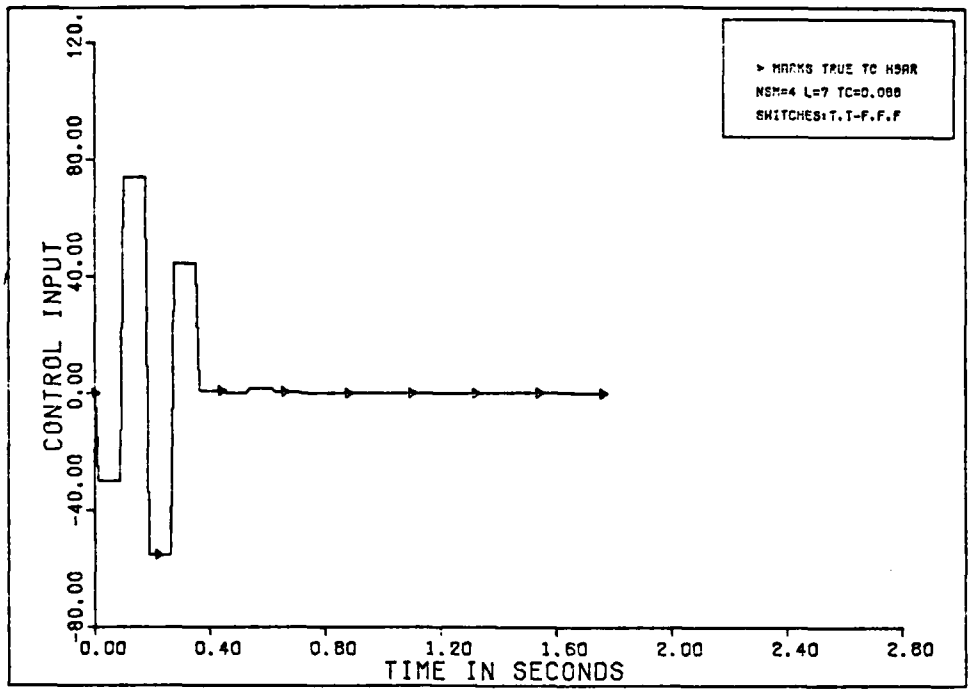


Fig C-29. Controls Applied for Truth Model to $\bar{H}$ Calculation, TC=0.088

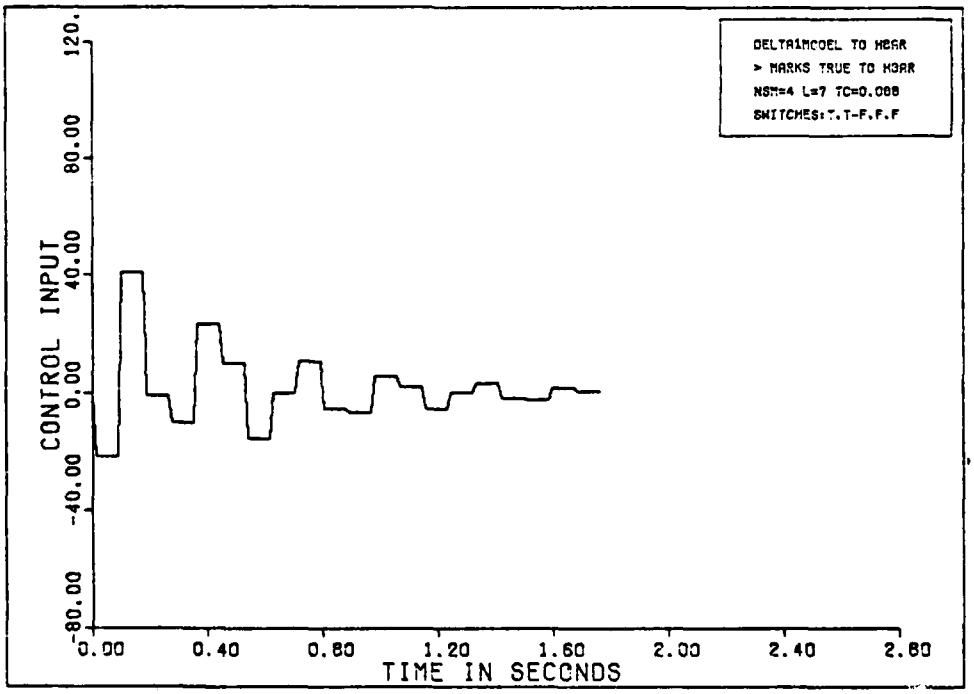


Fig C-30. Controls Applied for 10% Delta 1 Model to $\bar{H}$ Calculation, TC = 0.088

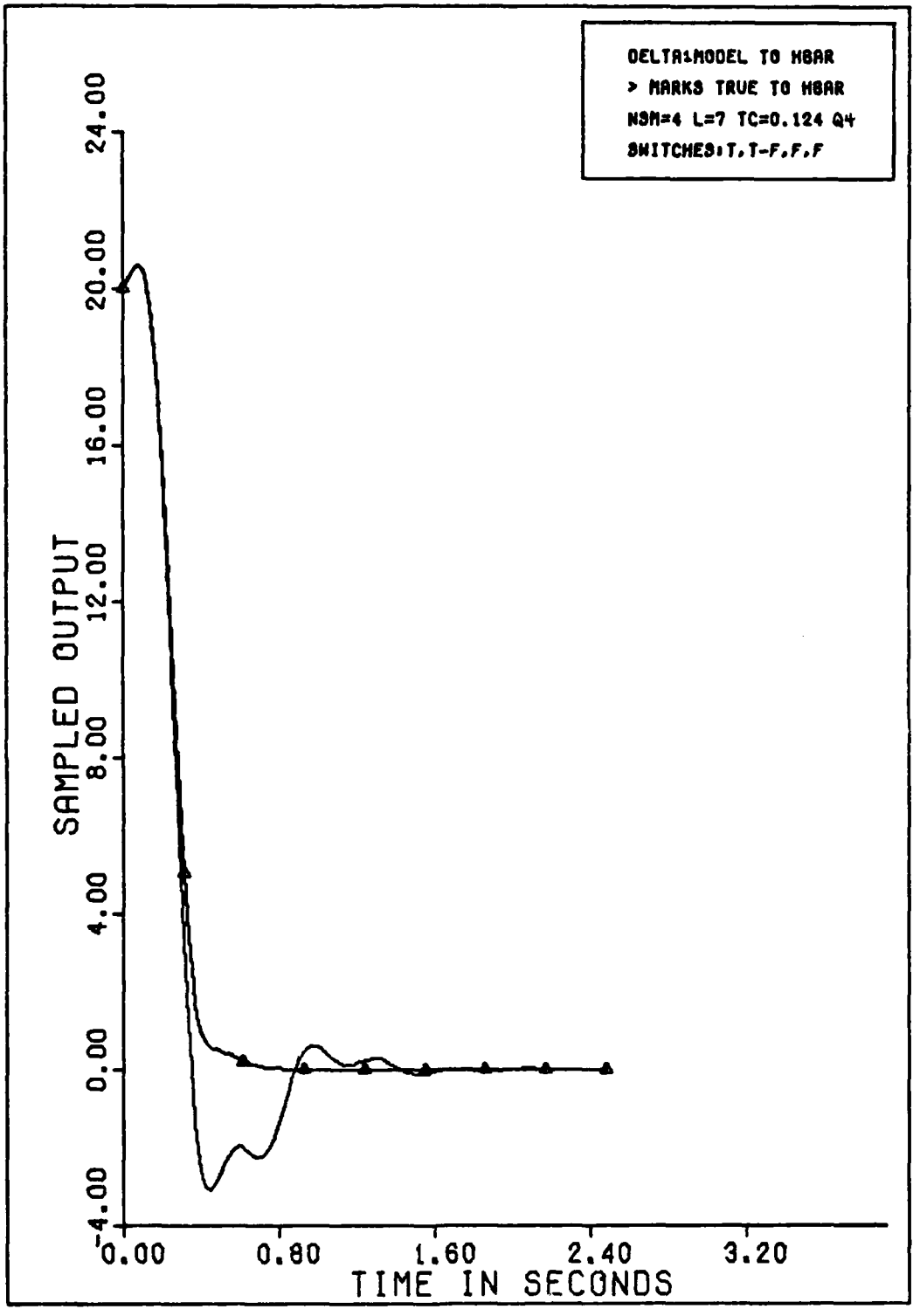


Fig C-31. System Output for Truth Model (Δ) and 10% Delta 1 Model to $\bar{H}$ Calculation, TC=0.124

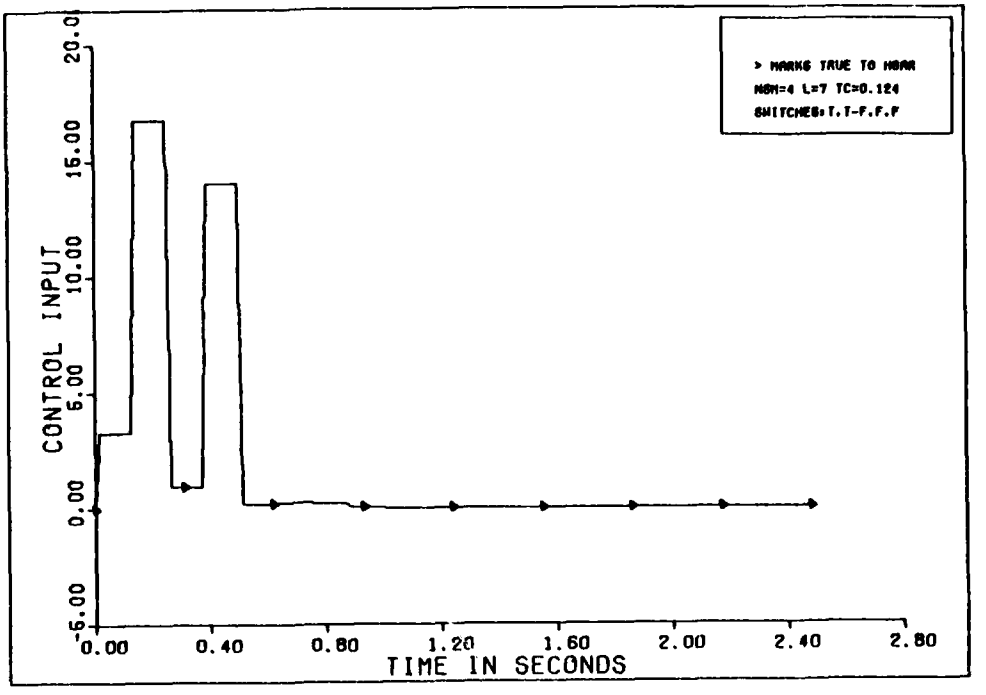


Fig C-32. Controls Applied for Truth Model to $\bar{H}$ Calculation, TC = 0.124

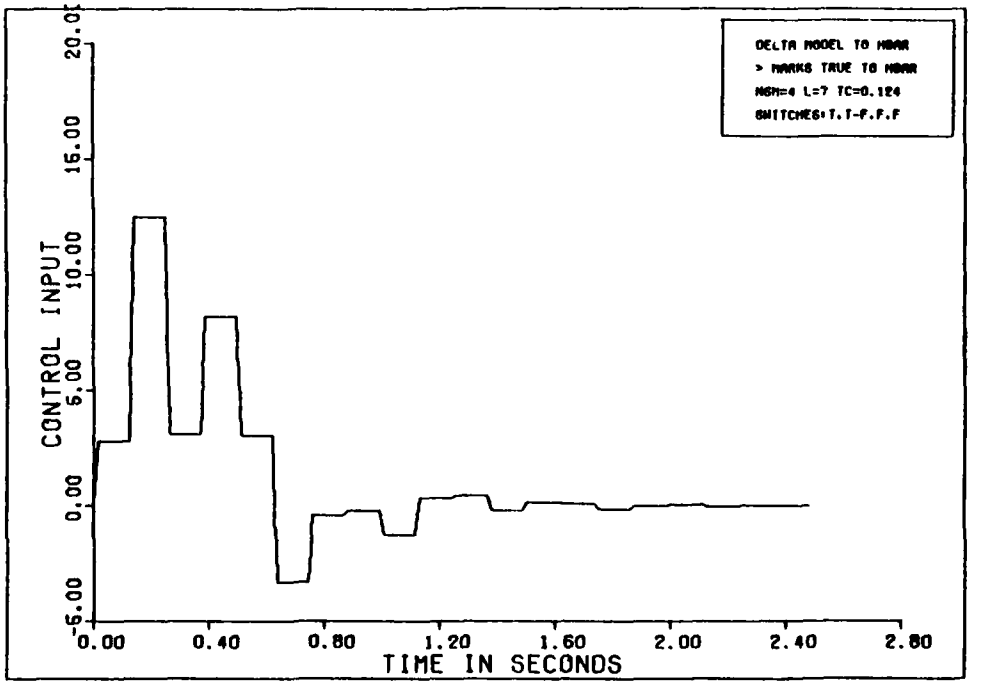


Fig C-33. Controls Applied for 10% Delta 1 Model to $\bar{H}$ Calculation, TC=0.124

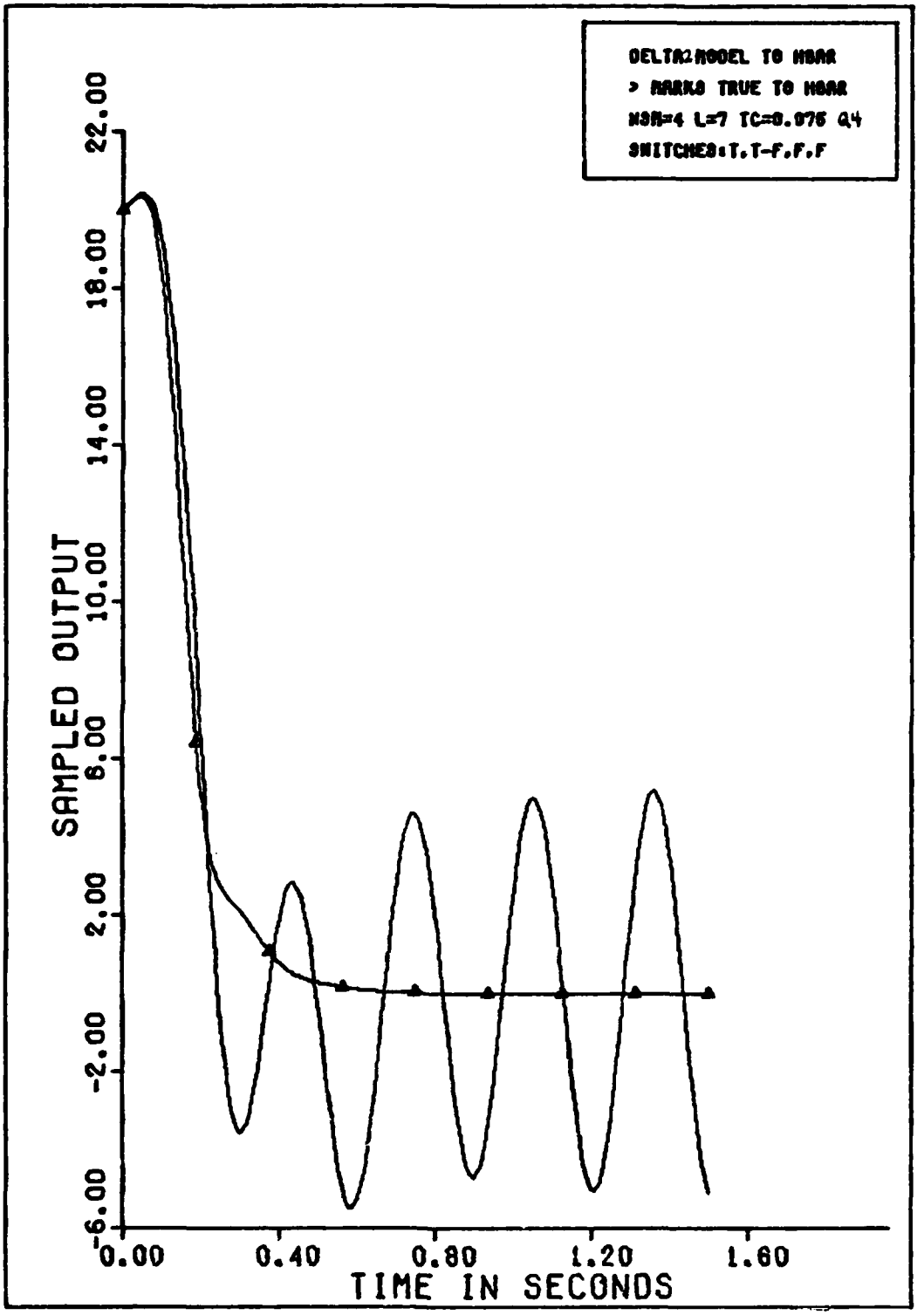


Fig C-34. System Output for Truth Model (Δ) and 20% Delta 2 Model to $\bar{H}$ Calculation, TC=0.075

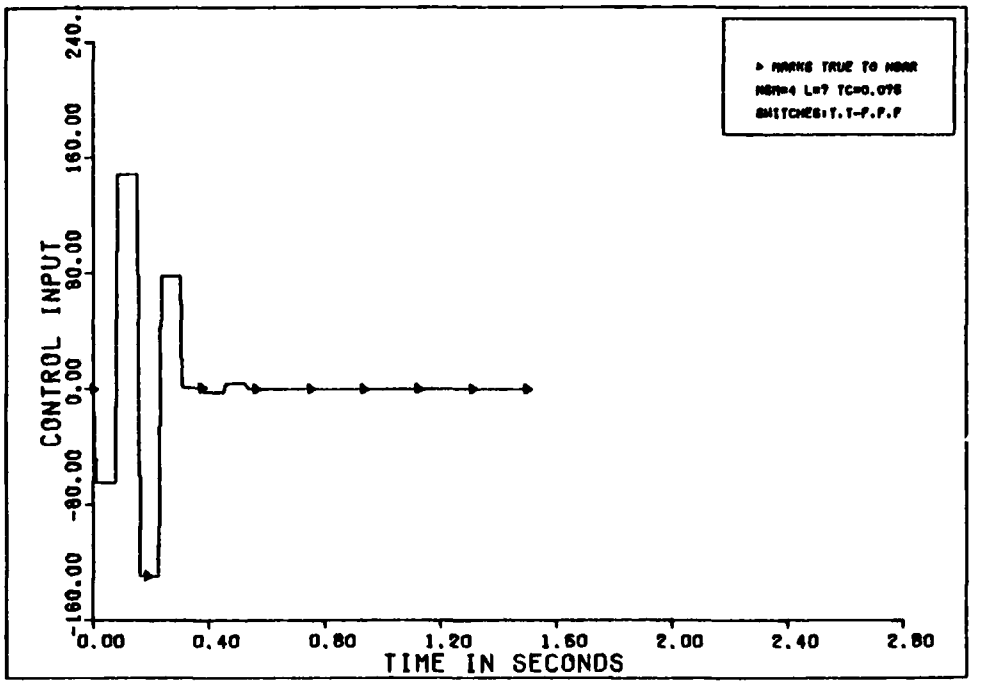


Fig C-35. Controls Applied for Truth Model to $\bar{H}$ Calculation, TC=0.075

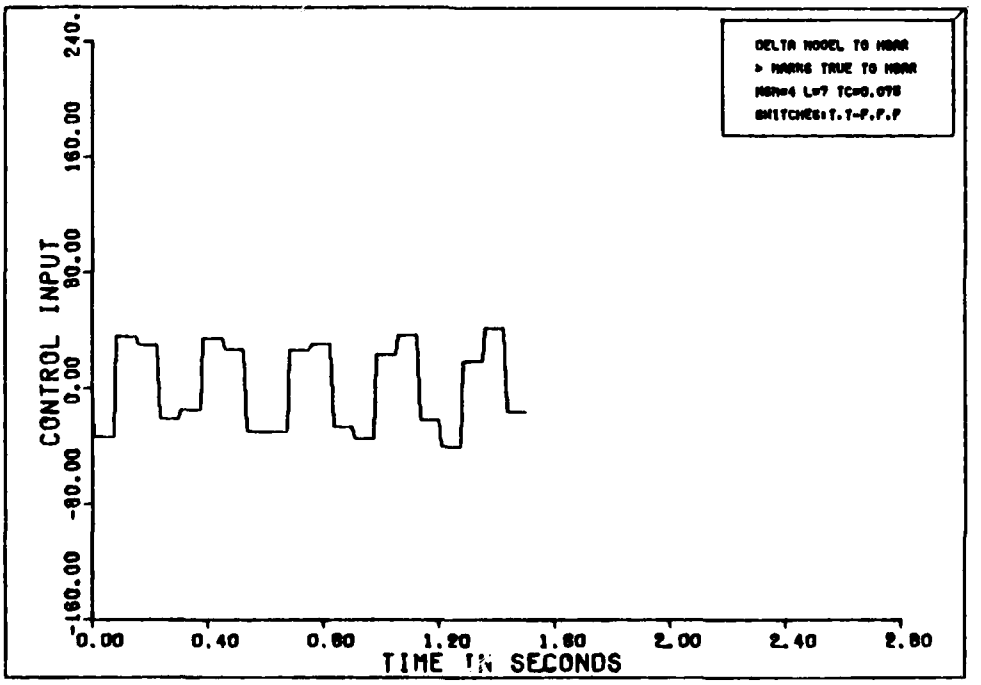


Fig C-36. Controls Applied for 20% Delta 2 Model to $\bar{H}$ Calculation, TC=0.075

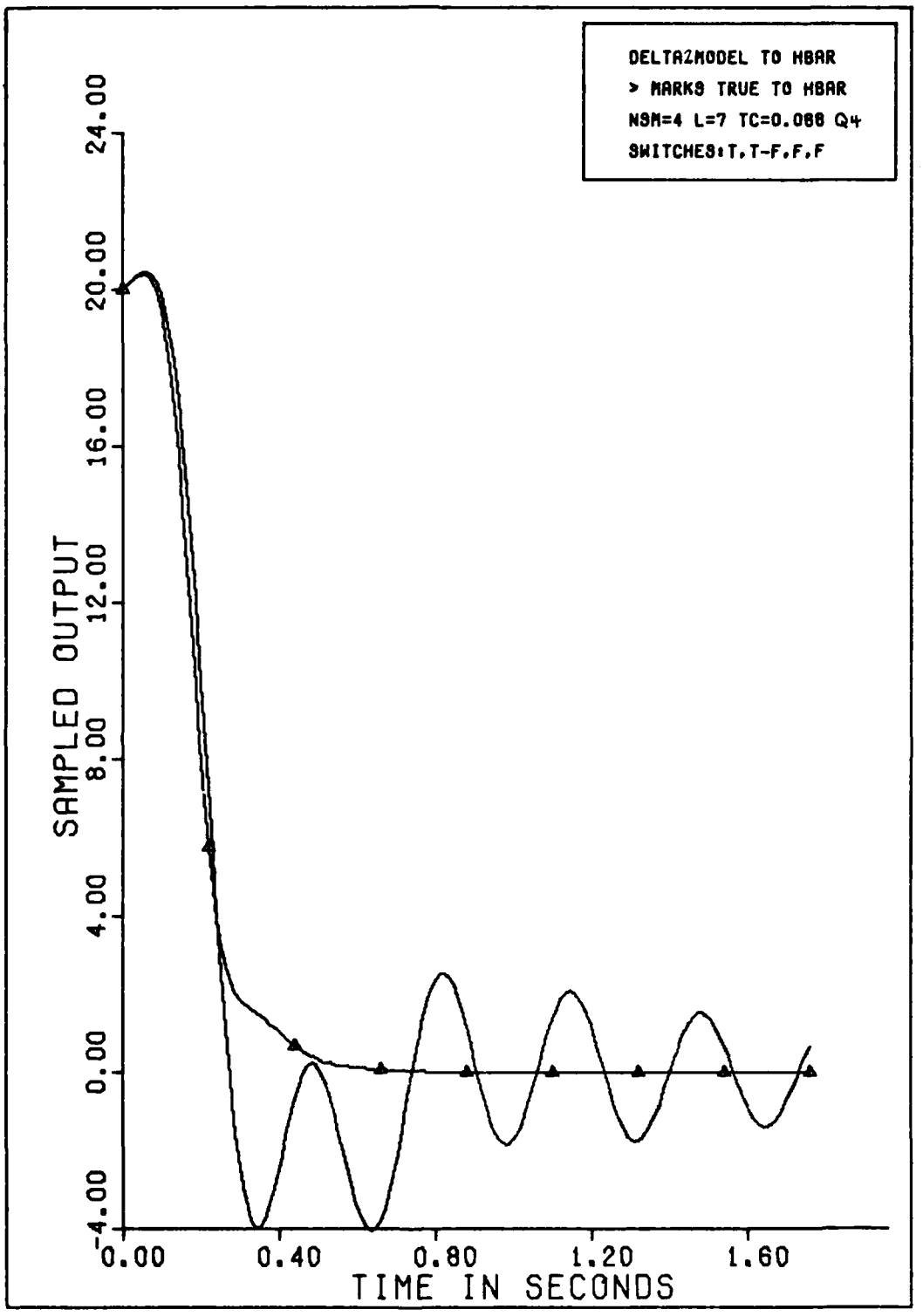


Fig C-37. System Output for Truth Model (Δ) and 20% Delta 2 Model to $\bar{H}$ Calculation, TC=0.088

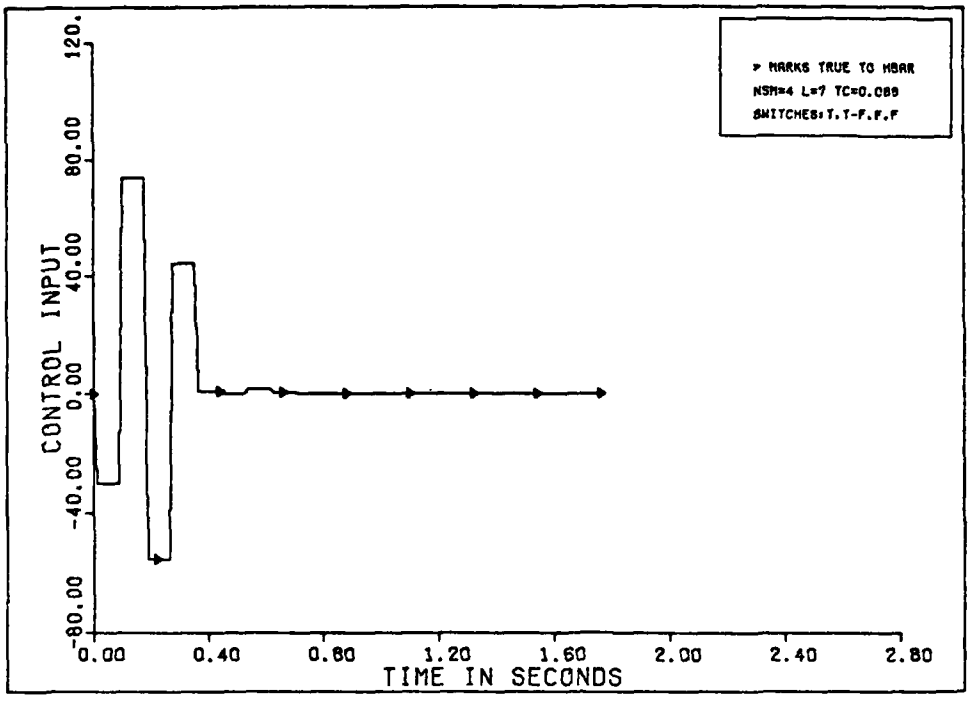


Fig C-38. Controls Applied for Truth Model to $\bar{H}$ Calculation, TC=0.088

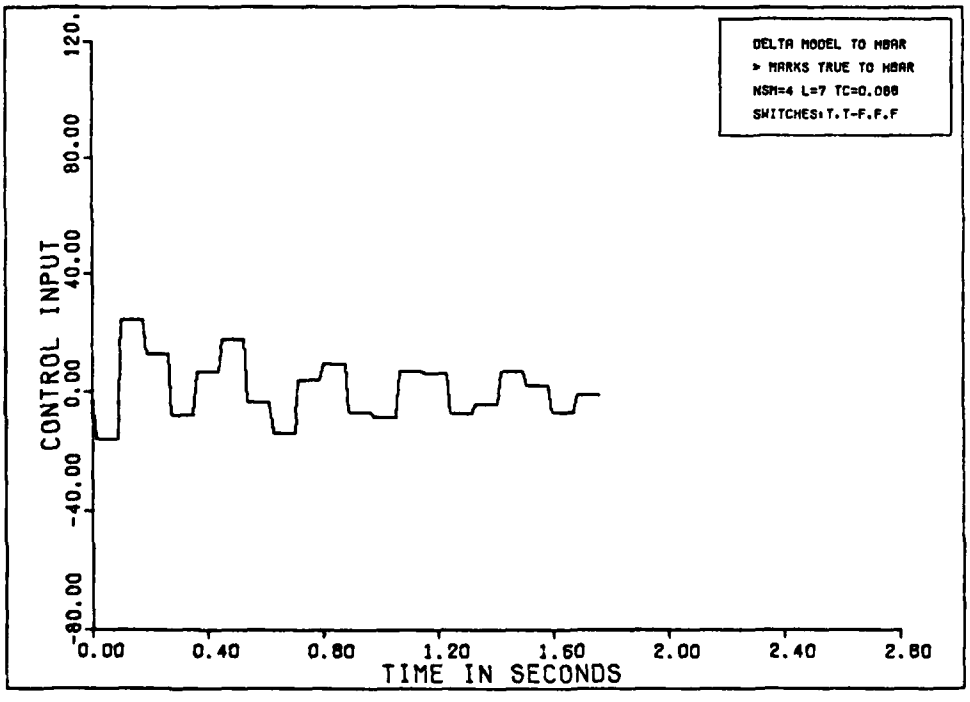


Fig C-39. Controls Applied for 20% Delta 2 Model to $\bar{H}$ Calculation, TC=0.088

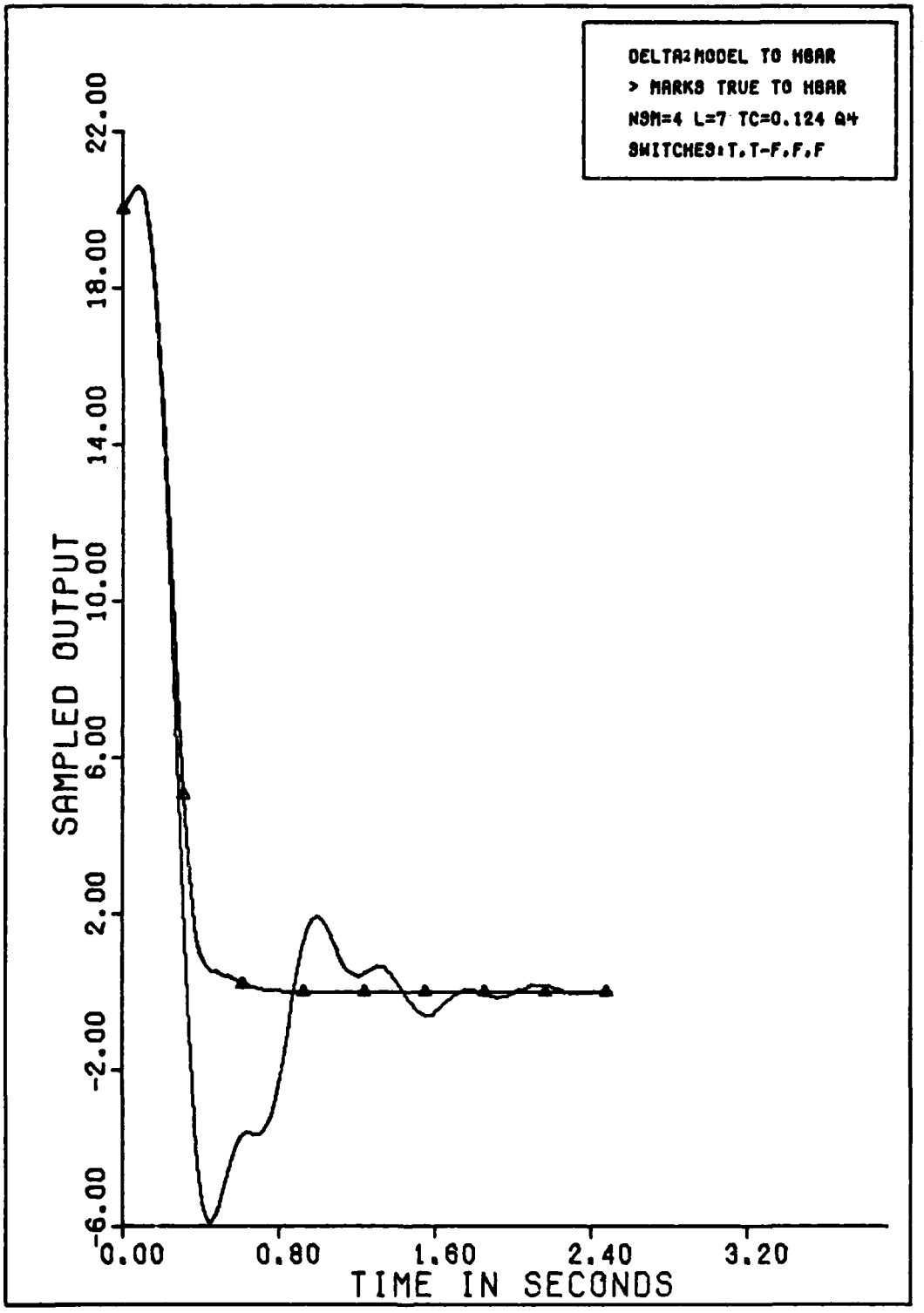


Fig C-40. System Output for Truth Model (Δ) and 20% Delta 2 Model to $\bar{H}$ Calculation, TC=0.124

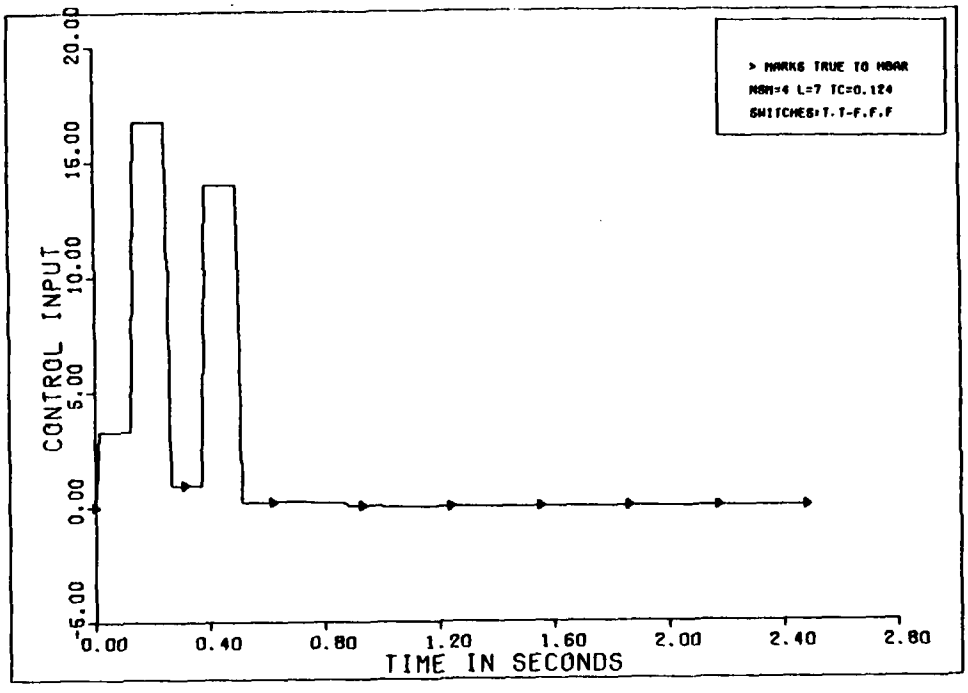


Fig C-41. Controls Applied for Truth Model to $\bar{H}$ Calculation, TC=0.124

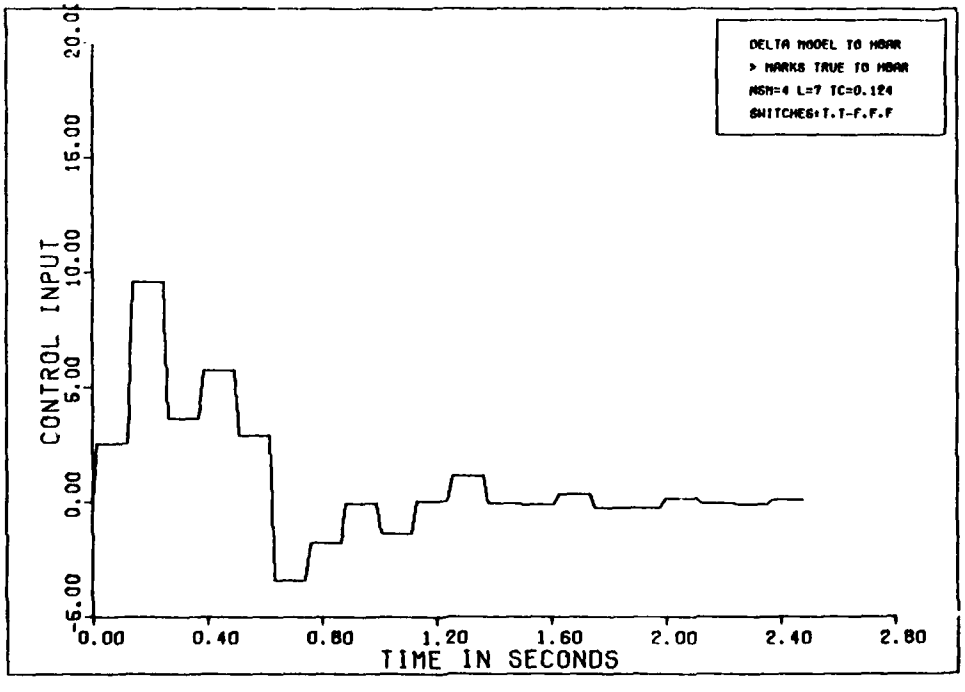


Fig C-42. Controls Applied for 20% Delta 2 Model to $\bar{H}$ Calculation, TC=0.124

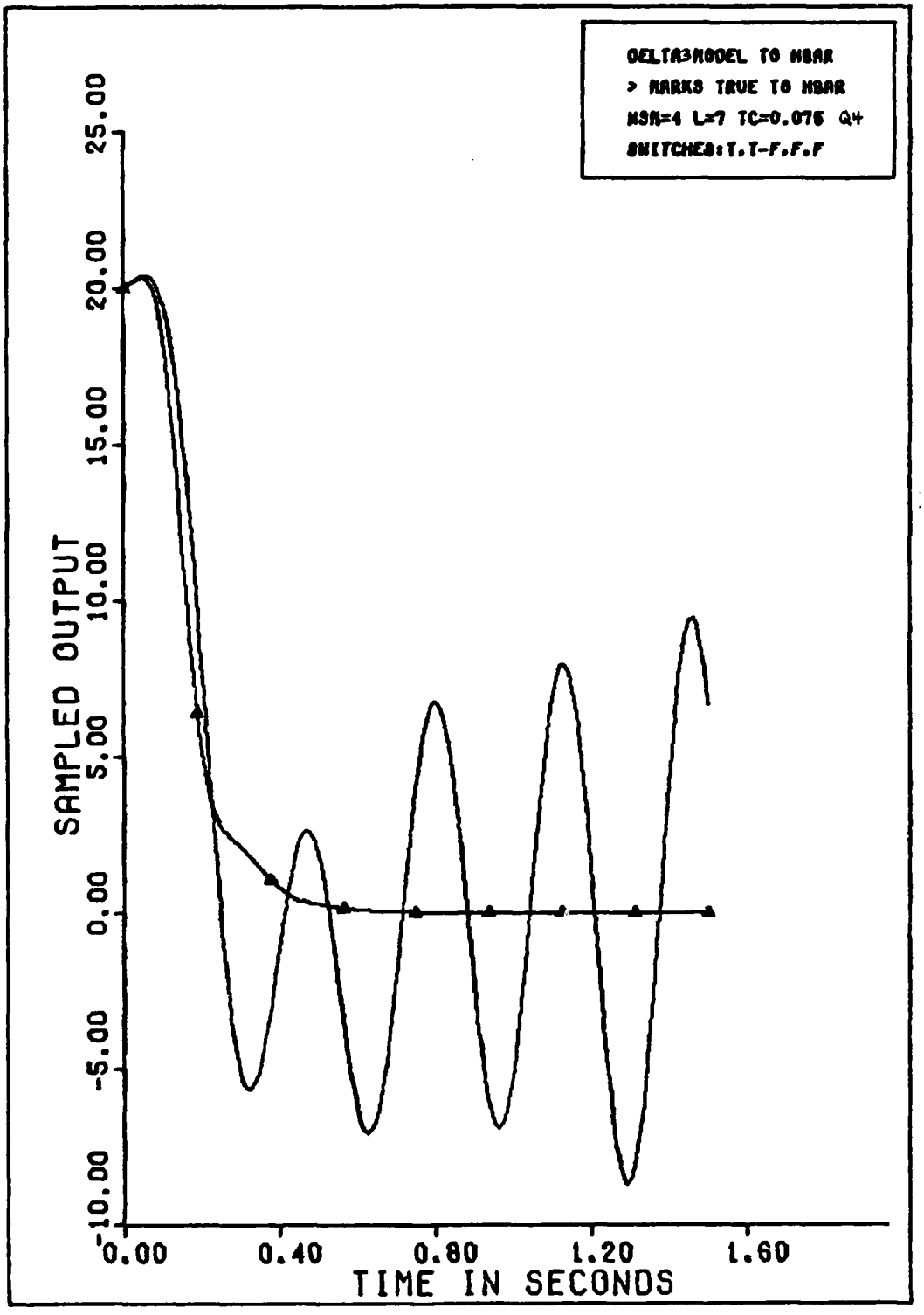


Fig C-43. System Output for Truth Model (Δ) and 30% Delta 3 Model to $\bar{H}$ Calculation, TC=0.075

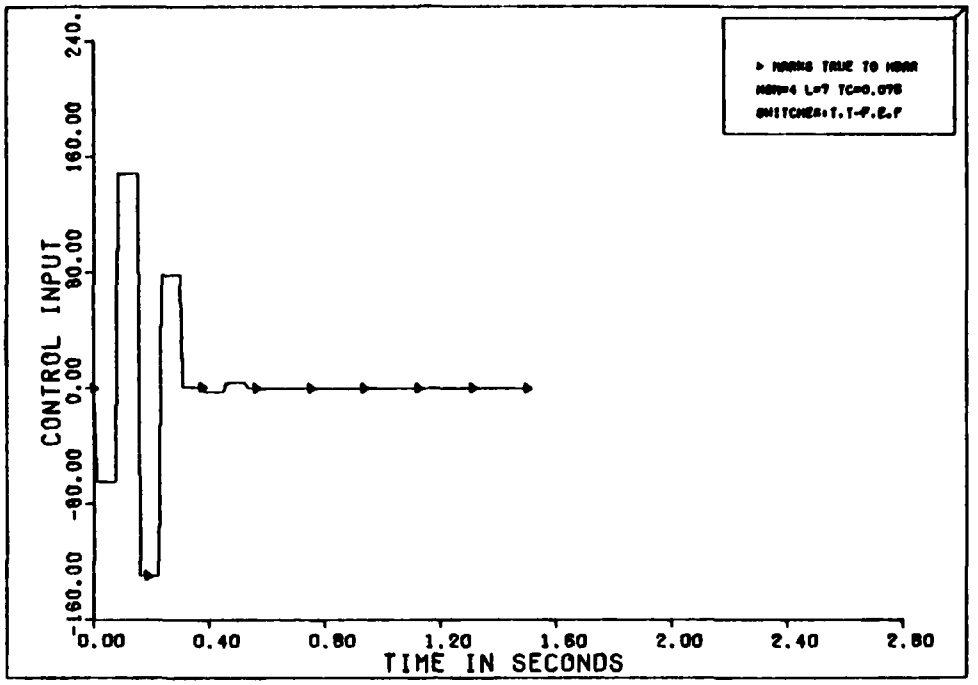


Fig C-44. Controls Applied for Truth Model to $\bar{H}$ Calculation, TC=0.075

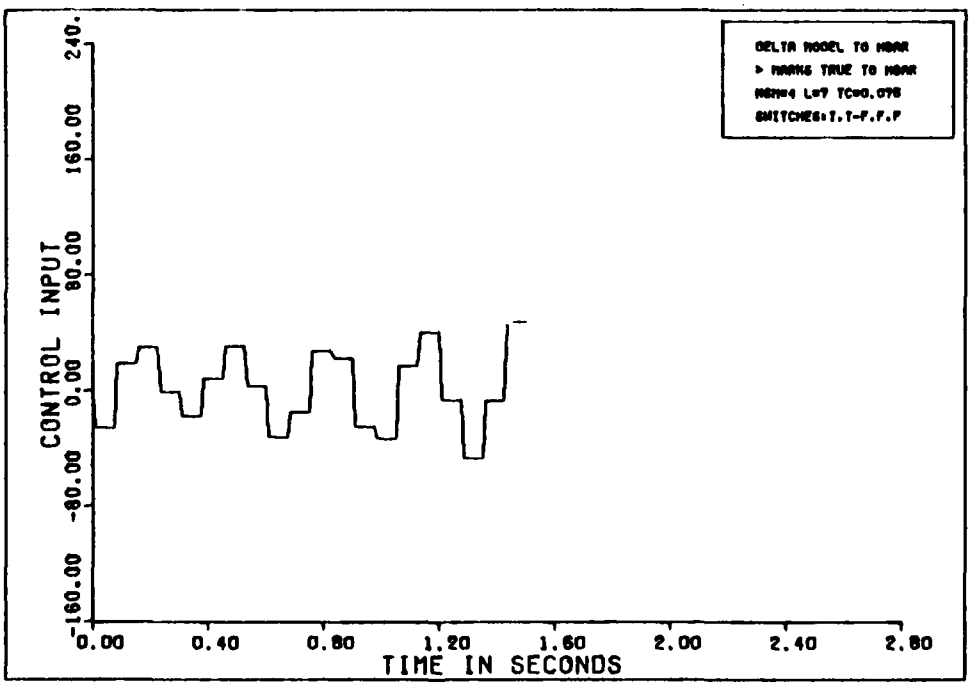


Fig C-45. Controls Applied for 30% Delta 3 Model to $\bar{H}$ Calculation, TC=0.075

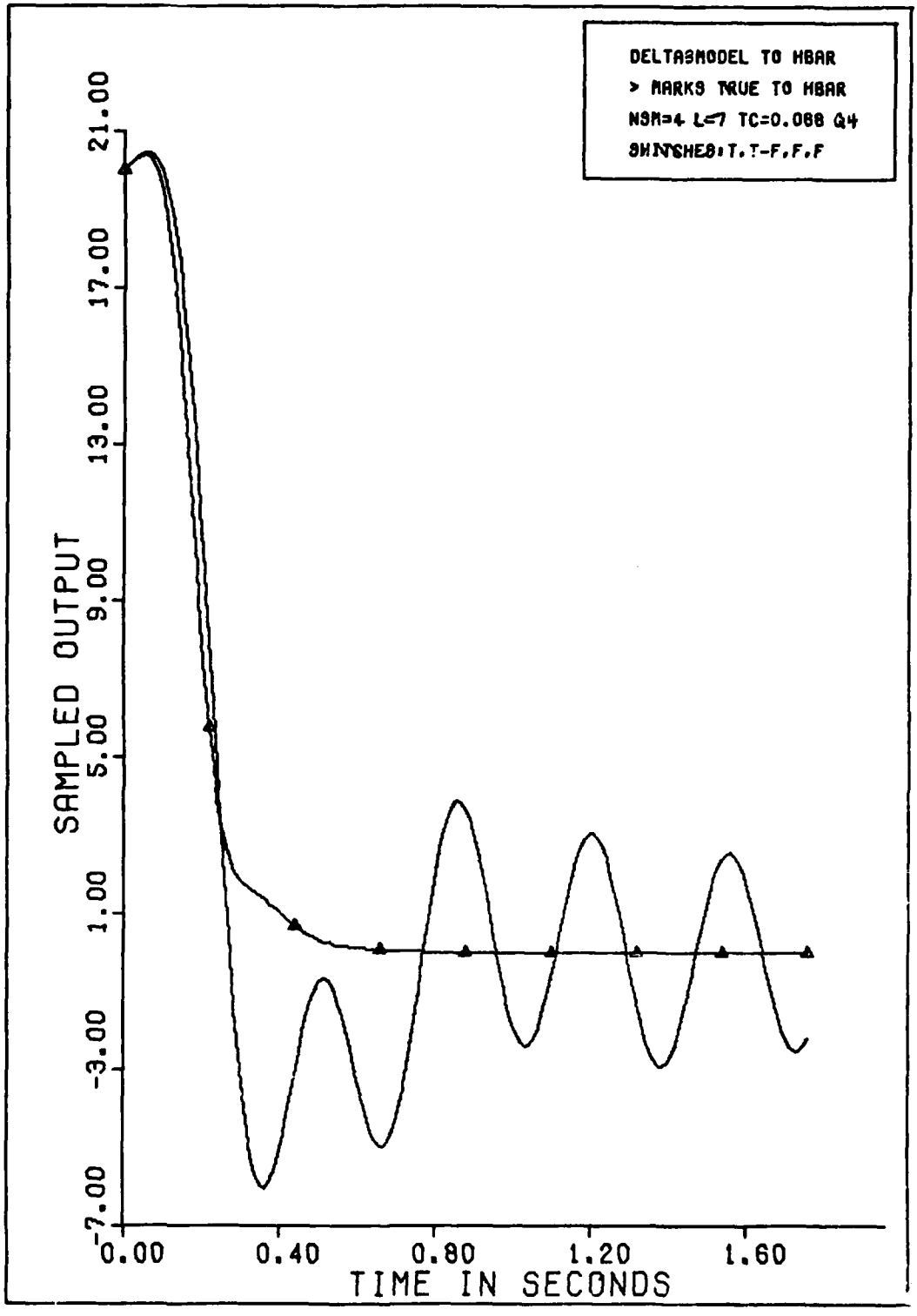


Fig C-46. System Output for Truth Model (Δ) and 30% Delta 3 Model to $\tilde{H}$ Calculation, TC=0.088

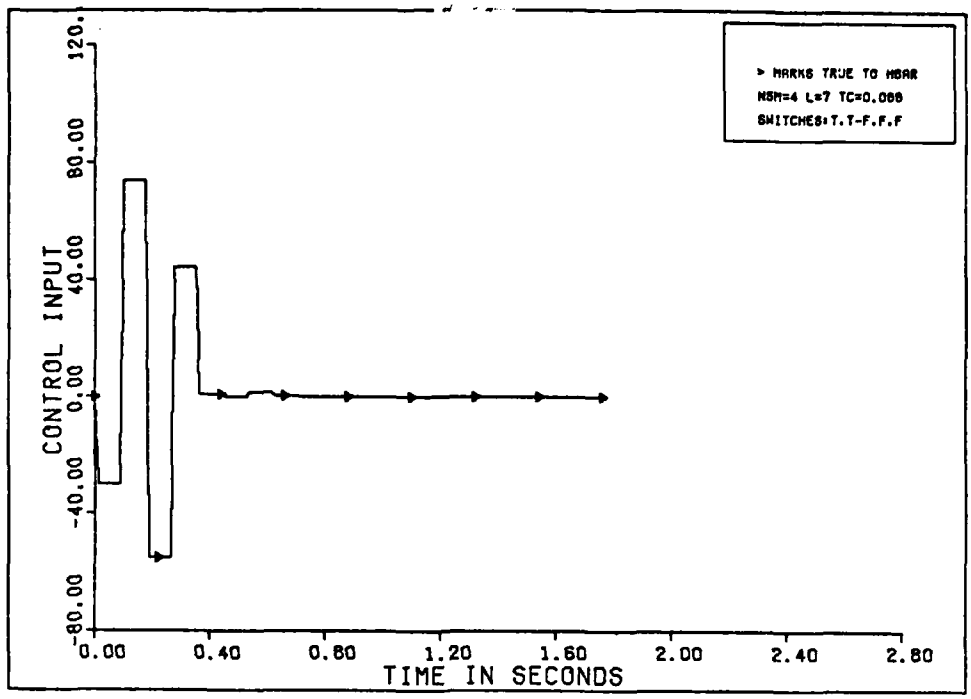


Fig C-47. Controls Applied for Truth Model to $\bar{H}$ Calculation, TC=0.088

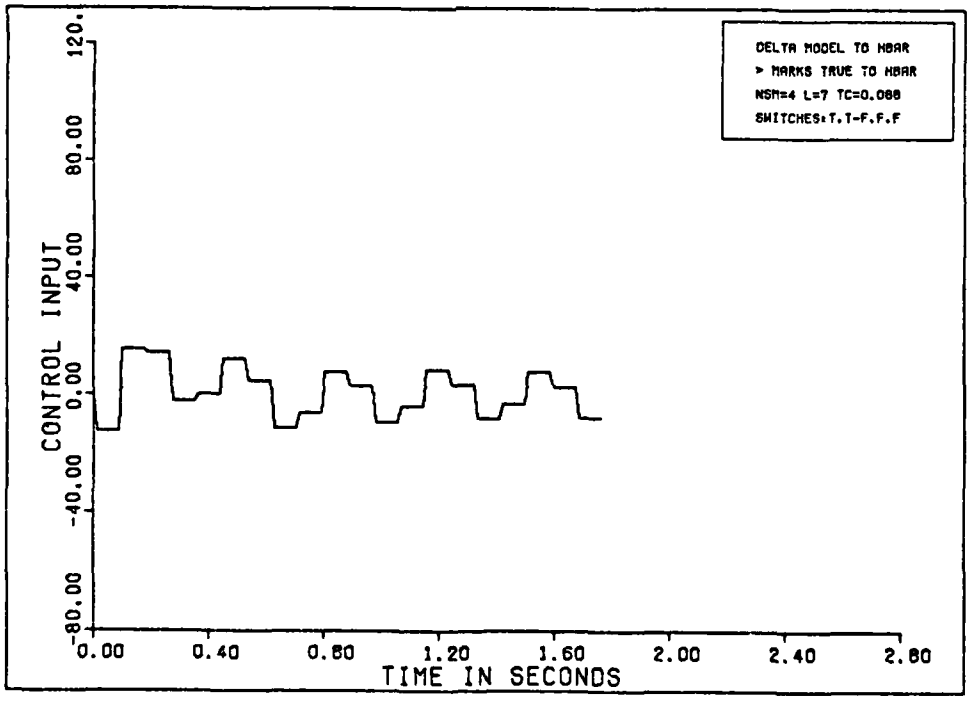


Fig C-48. Controls Applied for 30% Delta 3 Model to $\bar{H}$ Calculation, TC=0.088

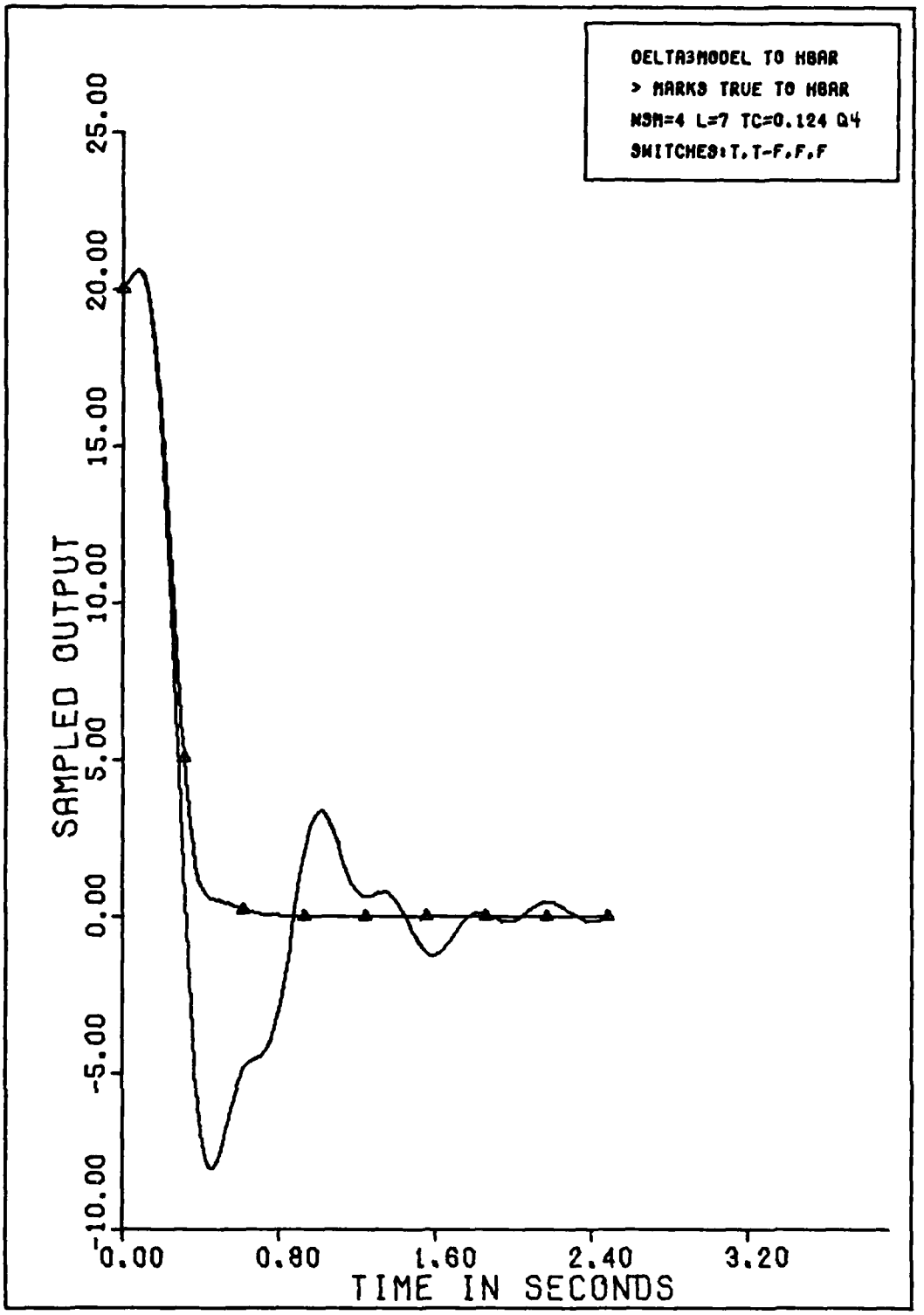


Fig C-49. System Output for Truth Model (Δ) and 30% Delta 3 Model to $\bar{H}$ Calculation, TC=0.124

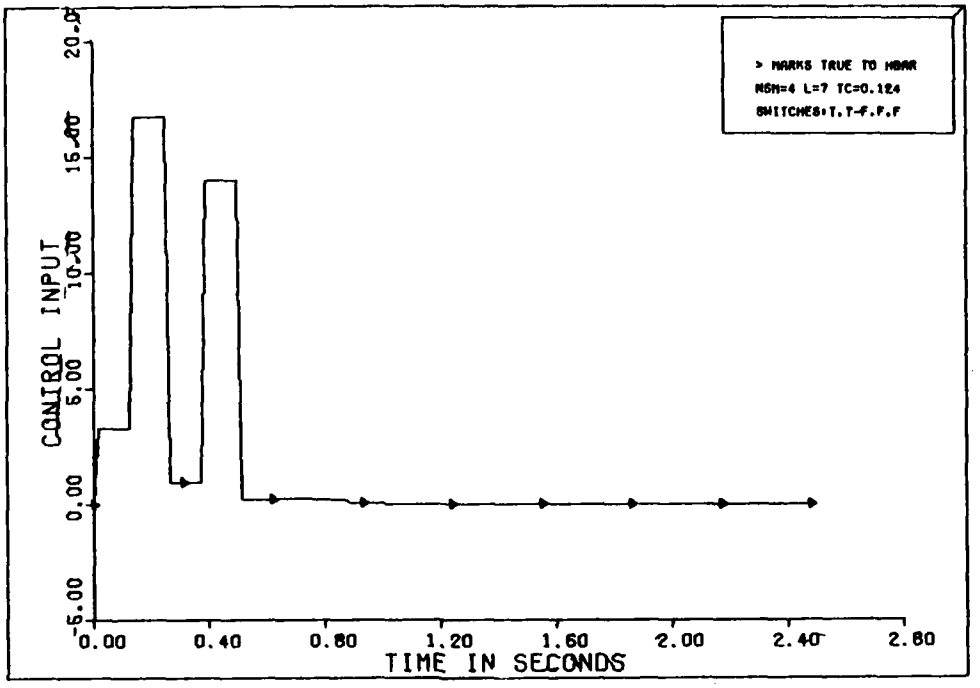


Fig C-50. Controls Applied for Truth Model to $\bar{H}$ Calculation, TC=0.124

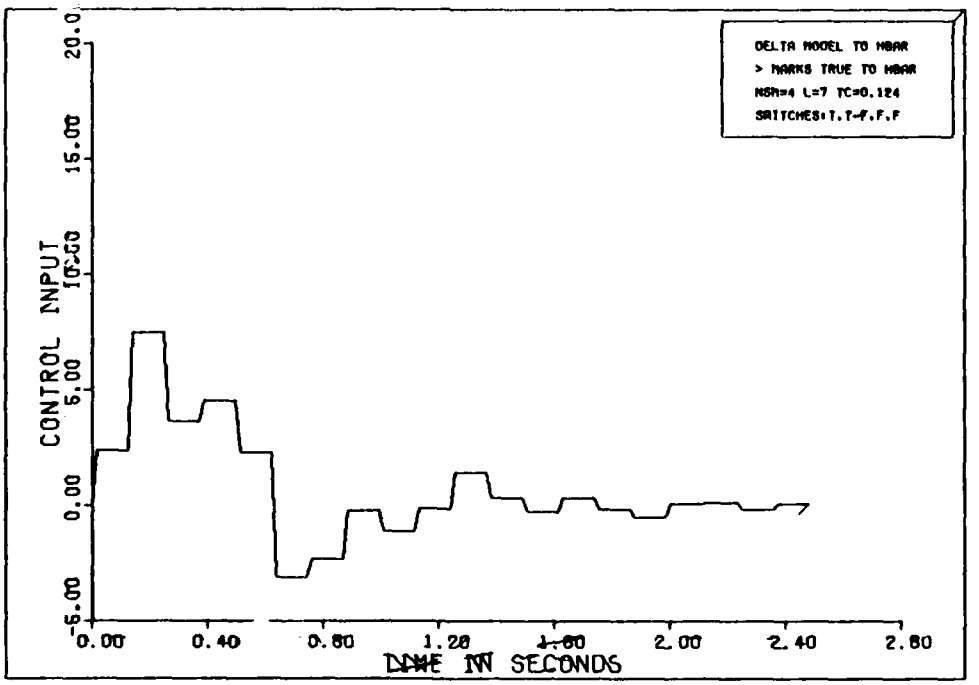


Fig C-51. Controls Applied for 30% Delta 3 Model to $\bar{H}$ Calculation, TC=0.124

Appendix D

FORTRAN Code for Terrain Following Test Cases of Chapter V

This Appendix contains the code used in the terrain following Test Case 1 of Chapter V. Two libraries must be attached prior to compilation:

1. IMSL, ID = Library, SN = ASD for a sorting routine.
2. CONTROL, ID = L720033, SN = AFML for a matrix inversion routine.

Figures 50 through 53 are outputs from this program.

```
      PROGRAM MOVIT(INPUT,OUTPUT,PLOT,TAPE5=INPUT,                      000100
     1TAPE6=OUTPUT)                                                     000110
C                                                                       000120
      DIMENSION COM1(20,20),AT(4,4),BT(4),CT(4),X(4),EAINT(4,4)         000130
      DIMENSION FLT(4,4),GLT(4),FST(4,4),GST(4)                         000140
      DIMENSION CFL(4),CFLK(4),HFC(28),HEC(4,7)                         000150
      DIMENSION QC(28),QEC(4,7),FSX(4),WK2(28)                          000160
      DIMENSION HTQH(7,7),HTQHI(7,7),HTQZ(7),UCOMP(7),GSU(4)            000170
      DIMENSION YC(28),YZ(28),ZC(28),ZEC(4,7),YSET(230)                 000180
      DIMENSION YS(603),US(603),TSS(603),IDYS(17),IDUS(17)              000190
      DIMENSION TSET(230),THETA(603),YST(631),IDTS(17),ERROR(202)       000200
      COMMON/MAIN1/NDIM,NDIM1,COM1/INOU/KIN,KOUT,KPUNCH                 000210
C                                                                       000220
C                                                                       000230
      EQUIVALENCE (HFC(1),HEC(1,1)),(QC(1),QEC(1,1))                    000240
      EQUIVALENCE (YS(1),YST(1)),(YSET(1),YST(602))                     000250
      EQUIVALENCE (ZC(1),ZEC(1,1))                                      000260
C  THE SYSTEM MODE ARE IN PHASE VBL FORM, B(N) IS                       000270
C  REALLY "1", THE ENTRY IN ITS PLACE IS A SCALE                        000280
C  FACTOR FOR THE INPUT. THIS DOES NOT AFFECT THE                       000290
C  OUTPUT.                                                              000300
C                                                                       000310
      DATA (AT(1,I),I=1,4)/-3.354,-12.24,0.,0./                         000320
      DATA (AT(2,I),I=1,4)/1.,-2.334,2*0./                              000330
      DATA (AT(3,I),I=1,4)/1.,3*0./                                     000340
      DATA (AT(4,I),I=1,4)/0.,-1077.8,1077.8,0./                        000350
      DATA (BT(I),I=1,4)/34.17,.331,0.,0./                              000360
      DATA (CT(I),I=1,4)/3*0.,1./                                       000370
      DATA (X(I),I=1,4)/4*0./                                           000380
      DATA NT,N,NFR1,NDR2/30,4,4,28/                                    000390
      DATA TC,L,NFR3,KF,NOTS,NSM/0.5,7,7*0,3,4/                         000400
      DATA TAU,NTC,A,W/.3,5E1,1000.,.5/                                 000410
      DATA IDYS(1)/2HSET(>) VS ACHIEVED H/                              000420
      DATA IDYS(3)/2HTC=.5 TAU=.3 NSM=4  /                              000430
      DATA IDYS(5)/2HL=7 A=1E3 W=.5,.6,.2/                              000440
      DATA IDYS(7)/2HA.C.=STKLEVEL  Q=03 /                              000450
      DATA IDYS(9)/2H   TIME IN SECONDS  /                              000460
      DATA IDYS(11)/2CHPROFILE ALTITUDE(FT)/                            000470
      DATA IDYS(13)/2CHFIG  . ACHIEVED AND DESIRED ALTITUDE AS A FN OF T000480
     1 /                                                                000490
```

```
C                                                                       000500
      DATA IOUS(1)/20HCONTROL INPUTS      /                             000510
      DATA IOUS(11)/20HELEVATOR CMD (RADS) /                            000520
      DATA IOUS(17)/50HFIG  . CONTROL INPUTS(RADS) AIRCRAFT#1           000530
     1 /                                                                000540
      DATA IOTS(1)/20HPITCH ANGLE (THETA) /                             000550
      DATA IOTS(11)/20HTHETA (RADIANS)    /                             000560
      DATA IOTS(17)/50HFG  . PITCH ANGLE                                000570
     + /                                                                000580
C                                                                       000590
C                                                                       000600
C                                                                       000610
      CALL PLOT(0.,-4.,-3)  $  CALL PLOT( .,0.03,-3)                    000620
      IDESIRE=TC/NSM                                                    000630
      TSET(1)=0.                                                        000640
      YSET(1)=0.                                                        000650
      DO 10 K=2,36                                                      000660
        YSET(K)=0.                                                      000670
        TSET(K)=TSET(K-1)+TDESIRE                                       000680
10    CONTINUE                                                          000690
      DO 20 K=37,228                                                    000700
        TSET(K)=TSET(K-1)+TDESIRE                                       000710
        KK=K-36                                                         000720
        YSET(K)=A*SIN(.5*TSET(KK))+A*SIN(.6*TSET(KK))+A*SIN(.2*TSET(KK)) 000730
20    CONTINUE                                                          000740
C                                                                       000750
      DO 45 I=1,17                                                      000760
        IOUS(I)=IOYS(I)                                                 000770
        IOTS(I)=IOYS(I)                                                 000780
45    CONTINUE                                                          000790
C  REINITIALIZE X AND B                                                 000800
C  WILL NOT HAVE TO WORRY ABOUT X IF USE RANDOM INITIAL STATES           000810
C  WHEN GO TO 10TH ORDER, SHOULD THINK ABOUT READING IN                 000820
      BT(1)=34.17                                                       000830
      BT(2)=.731                                                        000840
      BT(3)=0.                                                          000850
      BT(4)=0.                                                          000860
      DO 46 I=1,N                                                       000870
        X(I)=0.                                                         000880
46    CONTINUE                                                          000890
```

```
      MN=NSM*NOBS                                                       000900
      TOBS=TC/MN                                                        000910
      ALPHA=EXP(-TDESIRE/TAU)                                           000920
      ML=NSM*L                                                          000930
      KYSET=1                                                           000940
      KS=1                                                              000950
C  FORM Y(1)=CMX(0)                                                     000960
      Y=0.0                                                             000970
      DO 60 K=1,N                                                       000980
        Y=Y+CT(K) X(K)                                                  000990
        YS(1)=Y                                                         001000
60    CONTINUE                                                          001010
      US(1)=1.0                                                         001020
      THETA(1)=1.                                                       001030
      TSS(1)=0.0                                                        001040
      KIN=5                                                             001050
      KOUT=6                                                            001060
      KPUNCH=7                                                          001070
      CALL DISCRET(AT,BT,TDESIRE,TOBS,N,NT,NOR1,                        001080
     +EAINT,FLT,GLT,FST,GST)                                            001090
C                                                                       001100
      CALL WEIGHTS(NSM,L,ML,QC,QEQ)                                     001110
C                                                                       001120
      CALL APAR(CT,FLT,CFL,CFLK,GLT,HFC,HEQ,QEQ,HTQH,HTQHI,            001130
     +N,L,ML,NSM,NOR1)                                                  001140
C                                                                       001150
      DO 16  KC=1,KF                                                    001160
C  REMOVED HEADERS FOR DESIRED OUTPUT POINTS                            001170
C                                                                       001180
      CALL YDESIRE(Y,YSET(KYSET),ALPHA,ML,YD)                           001190
C                                                                       001200
C  SUBROUTINE YTI DESTROYS THE C MATRIX.  COPY C MATRIX INTO            001210
C  THE B MATRIX .                                                       001220
      DO 75 I=1,N                                                       001230
        BI(I)=CT(I)                                                     001240
75    CONTINUE                                                          001250
C                                                                       001260
      CALL YTI(X,N,ML,YZ,BT,FLT,CFL,NOR1)                               001270
C                                                                       001280
      CALL CONTROL(YD,YZ,ZC,ZEQ,HEQ,QEQ,HTQHI,HTQZ,UCOMP,              001290
```

```
     +U,L,ML,NSM,NOR3)                                                  001300
C  THESE ARE THE CONTROL LIMITS                                        001310
      IF(U.LE.-.262) U=-0.262                                          001320
      IF(U.GE.+.436) U=+0.436                                          001330
C                                                                      001340
C  NOW UPDATE STATES AND SAMPLE THE OUTPUT OF THE "ACTUAL" SYSTEM.     001350
C                                                                      001360
      KYSET=KYSET+NSM                                                  001370
      DO 15  KO=1,MN                                                   001380
        CALL STATE(X,FST,FSU,GST,GSU,N,U,NOR1)                         001390
C                                                                      001400
        CALL OUTPUT(X,CT,Y,N)                                          001410
        KS=KS+1                                                        001420
        US(KS)=U                                                       001430
      THETA(KS)=X(3)                                                   001440
        YS(KS)=Y                                                       001450
        TSS(KS)=TSS(KS-1)+TOBS                                         001460
150   CONTINUE                                                         001470
160   CONTINUE                                                         001480
      CALL SCALE(TSET,7.,229,1)                                        001490
      NTPT=NTP+228                                                     001500
      CALL SCALE(YST,5.,NTPT,1)                                        001510
      YSET1=YSET(1)                                                    001520
      YSET2=YSET(2)                                                    001530
      TSS(NTP+1)=TSET(229)                                             001540
      TSS(NTP+2)=TSET(230)                                             001550
      YS(NTP+1)=YST(NTPT+1)                                            001560
      YS(NTP+2)=YST(NTPT+2)                                            001570
      CALL HGRAPH(TSS,YS,NTP,IDYS,-1,0,0)                              001580
      YSET(1)=YSET1 & TSET201=TSET(201)                                001590
      YSET(2)=YSET2 & TSET202=TSET(202)                                001600
      CALL HGRAPH(TSET,YSET,228,IDYS,2,20,2)                           001610
      CALL HGRAPH(TSS,US,NTP,IDUS,1,0,0)                               001620
      CALL HGRAPH(TSS,THETA,NTP,IDTS,1,0,0)                            001630
      ERROR(1)=0.                                                      001640
      SUM=0.                                                           001650
      DO 17  K=2,200,1                                                 001660
        ERROR(K)=ABS(YSET(K)-YS(3*K-2))                                001670
        SUM=SUM+ERROR(K)                                               001680
170   CONTINUE                                                         001690
```

```
      AVE=SUM/200.                                                      001700
      CALL USMNMX(ERROR,200,1,XMIN,XMAX)                                001710
      PRINT ,"FOR W=",W,",PEAK ERROR=",XMAX,"AVG ERROR=",AVE            001720
      IDYS(1)="ALTITUDE  "                                              001730
      IDYS(2)="ERROR     "                                              001740
      IDYS(11)=IDYS(1)                                                  001750
      IDYS(12)="ERROR (FT)"                                             001760
      CALL HGRAPH(TSET,EPROR,200,IDYS,1,0,0)                            001770
      IDYS(2)="T. PROFILE"                                              001780
      IDYS(1)="DESIRED AL"                                              001790
      IDYS(11)="PROFILE AL"                                             001800
      IDYS(12)="ALTITUDE(FT)"                                           001810
      TSET(2,1)=TSET201                                                 001820
      TSET(2,2)=TSET202                                                 001830
      CALL HGRAPH(TSET,YSET,228,IDYS,1,0,0)                             001840
      CALL PLOTE(0)                                                     001850
      STOP "END OF MAIN"                                                001860
      END                                                               001870
C                                                                       001880
C                                                                       001890
      SUBROUTINE YDESIRE(Y,SET,ALPHA,ML,YD)                             001900
      DIMENSION YD(1),SET(1)                                            001910
C                                                                       001920
C  EQUATIONS PROGRAMED: YD(K)=YSET(K)-(ALPHA**K)(YSET-Y),               001930
C  WHERE ALPHA=EXP(-TDESIRE/TAU). TAU IS THE TIME CONSTANT              001940
C  OF THE FIRST ORDER DECAYING EXPONENTIAL.                             001950
C                                                                       001960
      SETNOW=SET(1)                                                     001970
      DO 10 K=1,ML                                                      001980
        YD(K)=SET(K)-((ALPHA**K)*(SETNOW-Y))                            001990
10    CONTINUE                                                          002000
      RETURN                                                            002010
      END                                                               002020
C                                                                       002030
C                                                                       002040
      SUBROUTINE YZI(X,N,ML,YZ,C,FL,CFL,NDR1)                           002050
      DIMENSION X(1),YZ(1),FL(NDR1,1),CFL(1),C(1)                       002060
C                                                                       002070
C  EQUATION PROGRAMED: YZ(K)=C*(FL**K)*X                                002080
C                                                                       002090
```

```
      DO 30 I=1,ML                                                      002100
        YZ(I)=0                                                         002110
        CALL RVMAT(C,FL,N,N,CFL,NDR1)                                   002120
        DO 10 J=1,N                                                     002130
          YZ(I)=YZ(I)+CFL(J)*X(J)                                       002140
10      CONTINUE                                                        002150
        DO 20 K=1,N                                                     002160
          C(K)=CFL(K)                                                   002170
20      CONTINUE                                                        002180
30    CONTINUE                                                          002190
      RETURN                                                            002200
      END                                                               002210
C                                                                       002220
C                                                                       002230
C                                                                       002240
      SUBROUTINE CONTROL(YD,YZ,ZC,ZEQ,HEQ,QEQ,HTQHI,HTQZ,UCOMP,        002250
     +U,L,ML,NSM,NDR3)                                                  002260
      DIMENSION YD(1),YZ(1),ZC(1),ZEQ(NSM,1),HEQ(NSM,1)                 002270
      DIMENSION QEQ(NSM,1),HTQHI(L,1),HTQZ(1),UCOMP(1)                  002280
C                                                                       002290
C  THE "NORMAL EQUATION" IS PROGRAMED AS FOLLOWS:                       002300
C  THE VECTOR OF INPUTS CALCULATED U=(HTQHI)*HTQZ.                      002310
C                                                                       002320
      DO 100 I=1,ML                                                     002330
      ZC(I)=YD(I)-YZ(I)                                                 002340
100   CONTINUE                                                          002350
C  FORM HTQZ                                                            002360
      DO 120 J=1,L                                                      002370
      HTQZ(J)=0.                                                        002380
      DO 110 K=J,L                                                      002390
      HTQZ(J)=DOTZ(HEQ(1,K-J+1),QEQ(1,K),ZEQ(1,K),NSM)+HTQZ(J)          002400
110   CONTINUE                                                          002410
120   CONTINUE                                                          002420
C  FORM UCOMP, THE CONTROL VECTOR                                       002430
      CALL MATV(HTQHI,HTQZ,UCOMP,L,L,7,7)                               002440
C  PICK OFF THE FIRST ELEMENT AS THE CONTROL TO                         002450
C  BE APPLIED                                                           002460
      U=UCOMP(1)                                                        002470
      RETURN                                                            002480
      END                                                               002490
```

```
C                                                                       002500
C                                                                       002510
C                                                                       002520
C                                                                       002530
      SUBROUTINE DISCRET(AT,BT,TDESIRE,TOBS,N,NT,NDR1,                  002540
     +EAINT,FLT,GLT,FST,GST)                                            002550
      DIMENSION AT(NDR1,1),BT(1),GST(1)                                 002560
      DIMENSION EAINT(NDR1,1),FLT(NDR1,1),GLT(1),FST(NDR1,1)            002570
      COMMON/MAIN1/NDIM,NDIM1,COM1/INOU/KIN,KOUT,KPUNCH                 002580
C                                                                       002590
C  THIS SUB DISCRETIZES THE A AND B MATRICES USING                      002600
C  ROUTINES ON A PACKAGE CALLED "CONTROL".                              002610
C  THE SAMPLE TIME OF TDESIRE IS HOW OFTEN THE CONTROL INPUT            002620
C  IS CALCULATED.  TIME TOBS IS THE DELTA T OF THE OUTPUT               002630
C  SAMPLER.                                                             002640
C                                                                       002650
      NDIM=NDR1                                                         002660
      NDIM1=NDR1+1                                                      002670
C  FOR THE "TRUTH" SYSTEM                                               002680
      CALL DSCRT(N,AT,TDESIRE,FLT,EAINT,NT)                             002690
      CALL MATV(EAINT,BT,GLT,N,N,NDR1,NDR1)                             002700
      CALL DSCRT(N,AT,TOBS,FST,EAINT,NT)                                002710
      CALL MATV(EAINT,BT,GST,N,N,NDR1,NDR1)                             002720
      RETURN                                                            002730
      END                                                               002740
C                                                                       002750
C                                                                       002760
      SUBROUTINE WEIGHT4(NSM,L,ML,QC,QEQ)                               002770
      DIMENSION QC(1),QEQ(NSM,L)                                        002780
C  THIS IS FOR MOD#2 GEO WTING                                          002790
      K=ML/2                                                            002800
      KK=K+1                                                            002810
      DO 10 I=1,K                                                       002820
        QC(I)=1.                                                        002830
10    CONTINUE                                                          002840
      DO 20 J=KK,ML                                                     002850
        QC(J)=1.0E+6                                                    002860
20    CONTINUE                                                          002870
      RETURN                                                            002880
      END                                                               002890
```

```
C                                                                       002900
C                                                                       002910
      SUBROUTINE STATE(X,FS,FSX,GS,GSU,N,U,NDR)                         002920
      DIMENSION X(1),FS(NDR,1),GS(1)                                    002930
      DIMENSION GSU(1),FSX(1)                                           002940
C                                                                       002950
C  EQUATION CODED: X(K+1)=FS*X(K)+GS*U, WHERE THE STATE MAY OR          002960
C  MAY NOT HAVE NOISE ON IT, DEPENDING ON THE SWITCH SETTINGS.          002970
C                                                                       002980
      DO 10 I=1,N                                                       002990
        GSU(I)=GS(I)*U                                                  003000
10    CONTINUE                                                          003010
      CALL MATV(FS,X,FSX,N,N,NDR,NDR)                                   003020
C REPLACE X WITH THE NEW STATE                                          003030
      DO 20 J=1,N                                                       003040
        X(J)=FSX(J)+GSU(J)                                              003050
20    CONTINUE                                                          003060
      RETURN                                                            003070
      END                                                               003080
C                                                                       003090
C                                                                       003100
      SUBROUTINE OUTPUT(X,C,Y,N)                                        003110
      DIMENSION X(1),C(1)                                               003120
C                                                                       003130
C  EQUATION CODED: Y=C*X, WHERE X MAY OR MAY NOT HAVE NOISE ON IT       003140
C                                                                       003150
      Y=0.0                                                             003160
      DO 10 I=1,N                                                       003170
        Y=Y+C(I)*X(I)                                                   003180
10    CONTINUE                                                          003190
      RETURN                                                            003200
      END                                                               003210
C                                                                       003220
C                                                                       003230
      SUBROUTINE PRINTM(NR,NC,DMAT,NDR)                                 003240
      DIMENSION DMAT(NDR,1)                                             003250
      DO 20 I=1,NR,1                                                    003260
        WRITE(5,10) I,(DMAT(I,J),J=1,NC,1)                              003270
10      FORMAT(1HO,I2,(T4,4(1PE13.4,1X)))                               003280
20    CONTINUE                                                          003290
```

```
      RETURN                                                            003300
      END                                                               003310
C                                                                       003320
C                                                                       003330
      SUBROUTINE PVMAT(RV,AMAT,NR,NC,VMAT,NDR)                          003340
      DIMENSION RV(1),AMAT(NDR,1),VMAT(1)                               003350
      DO 20 I=1,NC                                                      003360
        VMAT(I)=0.0                                                     003370
        DO 10 J=1,NC                                                    003380
          VMAT(I)=VMAT(I)+RV(J)*AMAT(J,I)                               003390
10      CONTINUE                                                        003400
20    CONTINUE                                                          003410
      RETURN                                                            003420
      END                                                               003430
C                                                                       003440
C                                                                       003450
      SUBROUTINE PRINTR(NE,RV,NDR)                                      003460
      DIMENSION RV(NDR)                                                 003470
      J=1                                                               003480
      WRITE 10,J,(RV(I),I=1,NE)                                         003490
10    FORMAT(1H0,I2,(T4,4(1PE13.4,1X)))                                 003500
      RETURN                                                            003510
      END                                                               003520
C                                                                       003530
C                                                                       003540
      SUBROUTINE MATV(AMAT,V,AMATV,NR,NC,NDR,NDC)                       003550
      DIMENSION AMAT(NDR,NDC),V(NDC),AMATV(NDR)                         003560
      DO 20 I=1,NC                                                      003570
        AMATV(I)=0.0                                                    003580
        DO 10 J=1,NC                                                    003590
          AMATV(I)=AMATV(I)+AMAT(I,J)*V(J)                              003600
10      CONTINUE                                                        003610
20    CONTINUE                                                          003620
      RETURN                                                            003630
      END                                                               003640
C                                                                       003650
C                                                                       003660
      SUBROUTINE HGRAPH(X,Y,N,ID,NO,NP,NS)                              003670
      DIMENSION X(1),Y(1),ID(1) $ IF(NO.EQ.2) GO TO 30                  003680
      IF (NO.LT.0) GO TO 10                                             003690
```

```
      CALL SCALE(X,7.,N,1)       $ CALL SCALE(Y,5.,N,1)                  003700
10    CALL PLOT(8.5,0.,-3) $   CALL PLOT( .,11.,3)                       003710
      CALL PLOT(-1.35,1.35,3)                                            003720
      CALL PLOT(-7.15,1.35,2) $  CALL PLOT(-7.15,9.65,2)                 003730
      IF(ID(1).EQ.000) GO TO 25                                          003740
      CALL PLOT(-7.05,9.55,3)     $ CALL PLOT(-7.05,7.55,2)              003750
      DO 20 I=1,7,2                                                      003760
20    CALL SYMBOL(7*.1-6.9,7.85 ,.07,ID(I),90.,20)                       003770
      CALL PLOT(-7.05,7.55,3)     $ CALL PLOT(-6.05,7.55,2)              003780
      CALL PLOT(-7.05,9.55,2)     $ CALL PLOT(-7.05,9.55,2)              003790
      CALL PLOT(-7.15,9.65,3)                                            003800
25    CALL PLOT(-1.35,9.65,2)   $ CALL PLOT(-1.35,1.35,2)                003810
      CALL SYMBOL(-6.65,1.15,.1,ID(13),0.,50)                            003820
      CALL AXIS(-1.85,2.1,ID(3),-20,7.,90.,X(N+1),X(N+2))                003830
      CALL AXIS(-1.85,2.1,ID(11),20,5.,10.,Y(N+1),Y(N+2))                003840
30    Y(N+2)=-Y(N+2)                                                     003850
      X(N+1)=X(N+1)-2.1*X(N+2) $ Y(N+1)=Y(N+1)+1.85*Y(N+2)               003860
      CALL LINE(Y,X,N,1,NP,NS)                                           003870
      X(N+1)=X(N+1)+2.1*X(N+2) $ Y(N+1)=Y(N+1)-1.85*Y(N+2)               003880
      Y(N+2)=-Y(N+2)                                                     003890
      RETURN                     $ END                                   003900
      SUBROUTINE VGRAPH(X,Y,N,ID,NO,NP,NS)                               003910
      DIMENSION X(1),Y(1),ID(1) $ IF(NO.EQ.2) GO TO 30                   003920
      IF (NO.LT.0) GO TO 10                                              003930
      CALL SCALE(X,4.9,N,1)       $ CALL SCALE(Y,7.0,N,1)                003940
10    CALL PLOT(8.5,0.,-3) $   CALL PLOT( .,11.,3)                       003950
      CALL PLOT(-1.35,1.35,3)                                            003960
      CALL PLOT(-7.15,1.35,2) $  CALL PLOT(-7.15,9.65,2)                 003970
      CALL PLOT(-1.35,9.65,2) $ IF(ID(1).EQ.000) GO TO 25                003980
      CALL PLOT(-1.45,9.55,3) $ CALL PLOT(-3.45,9.55,2)                  003990
      DO 20 I= 1,7,2                                                     004000
20    CALL SYMBOL(-3.15,9.7-I*.10,.07,ID(I),0.,20)                       004010
      CALL PLOT(-3.45,9.55,3) $ CALL PLOT(-3.45,8.55,2)                  004020
      CALL PLOT(-1.45,8.55,2) $ CALL PLOT(-1.45,9.55,2)                  004030
      CALL PLOT(-1.35,9.65,3)                                            004040
25    CALL PLOT(-1.35,1.35,2)                                            004050
      CALL SYMBOL(-6.65,1.15,.1,ID(13),0.,50)                            004060
      CALL AXIS(-1.85,1.85,ID(3),-20,4.9,0.,X(N+1),X(N+2))               004070
      CALL AXIS(-1.85,1.85,ID(11),20,7.0,90.,Y(N+1),Y(N+2))              004080
30    X(N+1)=X(N+1)+6.4*X(N+2)  $ Y(N+1)=Y(N+1)-1.85*Y(N+2)              004090
```

```
      CALL LINE(X,Y,N,1,NP,4S)                                          004100
      X(N+1)=X(N+1)-6.4*X(N+2)  $ Y(N+1)=Y(N+1)+1.85*Y(N+2)             004110
      RETURN      S END                                                 004120
C                                                                       004130
      FUNCTION DOT3(X,Q,Y,NSM)                                          004140
      DIMENSION X(1),Q(1),Y(1)                                          004150
      DOT3=0.                                                           004160
      DO 1 I=1,NSM                                                      004170
      DOT3=X(I)*Q(I)*Y(I)+DOT3                                          004180
1     CONTINUE                                                          004190
      RETURN                                                            004200
      END                                                               004210
C                                                                       004220
C                                                                       004230
      SUBROUTINE HBAR(C,FL,CFL,CFLK,GL,HFC,HEQ,QEQ,HTQH,HTQHI,          004240
     +N,L,ML,NSM,NDR1)                                                  004250
      DIMENSION C(1),FL(NDR1,1),GL(1),CFL(1),CFLK(1)                    004260
      DIMENSION HFC(1),HEQ(NSM,1),QEQ(NSM,1),HTQH(L,1),HTQHI(L,1)       004270
      COMMON/MAINS/NDIM,NDIM1,COM1/INOU/KIN,KOUT,KPUNCH                 004280
C                                                                       004290
C                                                                       004300
C                                                                       004310
C  NOW GET THE PIECES OF THE "NORMAL EQN", WHICH                        004320
C  IS USED TO CALCULATE THE CONTROL USED TO DRIVE                       004330
C  THE OUTPUT ALONG THE DESIRED TRAJECTORY.                             004340
C                                                                       004350
C                                                                       004360
C  THIS BLOCK(THRU SN40) CREATES AN EQUIVALENT OF HBAR                  004370
C  WHEN GO TO MIMC JUST CALL RVPAT                                      004380
C  THE ELEMENTS OF THE FIRST COL OF HBAR,HFC, IS FORMED USING THE       004390
C  RELATIONSHIP HFC(I)=SUM OVER I OF(C*F** (I-1)*G).                    004400
C                                                                       004410
C                                                                       004420
      NDIM=L                                                            004430
      NDIM1=L+1                                                         004440
C  CHANGE L TO NDR3 WHEN GO TO 10TH ORDER                               004450
      HFC(1)=0                                                          004460
      DO 10 I=1,N                                                       004470
        HFC(1)=HFC(1)+C(I)*GL(I)                                        004480
10    CONTINUE                                                          004490
```

```
C  FORM CFL                                                              004500
      CALL RVMAT(C,FL,N,N,CFL,NDR1)                                      004510
      DO 40 J=2,ML                                                       004520
        CFLGL=0.0                                                        004530
        DO 20 JJ=1,N                                                     004540
          CFLGL=CFLGL+CFL(JJ)*GL(JJ)                                     004550
20      CONTINUE                                                         004560
        HFC(J)=HFC(J-1)+CFLGL                                            004570
        CALL RVMAT(CFL,FL,N,N,CFLK,NDP1)                                 004580
C  EQUATE THE TWO MATRICES SO CAN GO THRU INNER                          004590
C  LOOP AGAIN                                                            004600
        DO 30 II=1,N                                                     004610
          CFL(II)=CFLK(II)                                               004620
30      CONTINUE                                                         004630
40    CONTINUE                                                           004640
C                                                                        004650
C                                                                        004660
C  WILL NOW FORM THE REST OF THE PIECES NEEDED FOR THE                   004670
C  'NORMAL' EQN USED IN CALCULATION OF THE CONTROL INPUT.                004680
C  FORM HTQH                                                             004690
C  THIS IS THE RETD METHOD                                               004700
      DO 70 I=1,L                                                        004710
      DO 60 J=I,L                                                        004720
      HTQH(I,J)=0.                                                       004730
      DO 50 K=J,L                                                        004740
      HTQH(I,J)=DOT3(HEQ(1,(-I+1),OF2(1,K),HEQ(1,K-J+1),NSM)+HTQH(I,J)   004750
50    CONTINUE                                                           004760
60    CONTINUE                                                           004770
70    CONTINUE                                                           004780
C  HAVE FORMED THE UPPER SECTION OF HEQ,OFF DIAGONAL TERMS               004790
C  ARE TRANSPOSES.  IN THE SISO CASE ELEMENTS ARE SCALARS                004800
C  SO THE OFF DIAGONAL TERMS ARE EQUAL.  THIS BLOCK                      004810
C  DEFINES THE LOWER ELEMENTS IN TERMS OF THE UPPER ONES.                004820
      DO 90 I=1,L                                                        004830
      DO 80 J=I,L                                                        004840
      IF(I.EQ.J) GO TO 80                                                004850
      HTQH(J,I)=HTQH(I,J)                                                004860
80    CONTINUE                                                           004870
90    CONTINUE                                                           004880
      CALL SMINV(L,L,HTQH,HTQHI,MR,1)                                    004890
```

```
      RETURN                                                            004900
      END                                                               004910
C                                                                       004920
C                                                                       004930
C                                                                       004940
      SUBROUTINE WEIGHT1(NSM,L,ML,QC,QEQ)                               004950
      DIMENSION QC(1),QEQ(NSM,L)                                        004960
C  THIS IS FOR IDENTITY WEIGHTING                                       004970
      DO 10 I=1,ML                                                      004980
        QC(I)=1.0                                                       004990
10    CONTINUE                                                          005000
      RETURN                                                            005010
      END                                                               005020
C                                                                       005030
C                                                                       005040
      SUBROUTINE WEIGHT2(NSM,L,ML,QC,QEQ)                               005050
      DIMENSION QC(1),QEQ(NSM,L)                                        005060
C  THIS IS FOR BLOCK WEIGHTING                                          005070
      QEQ(1,1)=1.                                                       005080
      DO 10 I=2,L                                                       005090
        QEQ(1,I)=2*QEQ(1,I-1)                                           005100
10    CONTINUE                                                          005110
      DO 30 I=1,L                                                       005120
        DO 20 J=2,NSM                                                   005130
          QEQ(J,I)=QEQ(1,I)                                             005140
20      CONTINUE                                                        005150
30    CONTINUE                                                          005160
      RETURN                                                            005170
      END                                                               005180
C                                                                       005190
C                                                                       005200
      SUBROUTINE WEIGHT3(NSM,L,ML,QC,QEQ)                               005210
      DIMENSION QC(1),QEQ(NSM,L)                                        005220
C  THIS IS FOR GEO WTING                                                005230
      QC(1)=1.                                                          005240
      DO 10 I=2,ML                                                      005250
        QC(I)=2*QC(I-1)                                                 005260
10    CONTINUE                                                          005270
      RETURN                                                            005280
      END                                                               005290
```

```
C                                                                       005300
C                                                                       005310
      SUBROUTINE WEIGHT5(NSM,L,ML,QC,QEQ)                               005320
      DIMENSION QC(1),QEQ(NSM,L)                                        005330
C  THIS IS FOR EARLY GEO WTING                                          005340
      QC(ML)=1.                                                         005350
      DO 10 IB=2,ML                                                     005360
        I=ML+1-IB                                                       005370
        QC(I)=2*QC(I+1)                                                 005380
10    CONTINUE                                                          005390
      RETURN                                                            005400
      END                                                               005410
C                                                                       005420
C                                                                       005430
      SUBROUTINE WEIGHT6(NSM,L,ML,QC,QEQ)                               005440
      DIMENSION QC(1),QEQ(NSM,L)                                        005450
C                                                                       005460
C  THIS IS FOR REVERSE BLOCK                                            005470
      QEQ(1,L)=1.                                                       005480
      DO 10 IB=2,L                                                      005490
        I=L+1-IB                                                        005500
        QEQ(1,I)=2*QEQ(1,I+1)                                           005510
10    CONTINUE                                                          005520
      DO 30 I=1,L                                                       005530
        DO 20 J=2,NSM                                                   005540
          QEQ(J,I)=QEQ(1,I)                                             005550
20      CONTINUE                                                        005560
30    CONTINUE                                                          005570
      RETURN                                                            005580
      END                                                               005590
```

```
      PROGRAM MOVIT(INPUT,OUTPUT,PLOT,TAPE5=INPUT,                      000100
     1TAPE6=OUTPUT)                                                     000110
C                                                                       000120
      DIMENSION COM1(20,23),AT(4,4),BT(4),CT(4),X(4),EAINT(4,4)         000130
      DIMENSION FLT(4,4),GLT(4),FST(4,4),GST(4)                         000140
      DIMENSION CFL(4),CFLK(4),HFC(28),HEC(4,7)                         000150
      DIMENSION QC(28),QEC(4,7),FSX(4),WK(28)                           000160
      DIMENSION HTQH(7,7),HTQHI(7,7),HTQZ(7),UCOMP(7),GSU(4)            000170
      DIMENSION YC(28),YZ(23),ZC(28),ZEC(4,7),YSET(230)                 000180
      DIMENSION YS(603),US(603),TSS(603),IDYS(17),IDUS(17)              000190
      DIMENSION TSET(230),THETA(603),YST(631),IDTS(17),ERROR(202)       000200
      COMMON/MAIN1/NDIM,NDTM1,COM1/INOU/KIN,KOUT,KPUNCH                 000210
C                                                                       000220
C                                                                       000230
      EQUIVALENCE (HFC(1),HEC(1,1)),(QC(1),QEC(1,1))                    000240
      EQUIVALENCE (YS(1),YST(1)),(YSET(1),YST(602))                     000250
      EQUIVALENCE (ZC(1),ZEC(1,1))                                      000260
C  THE SYSTEM CONS ARE IN PHASE VBL FORM, B(N) IS                       000270
C  REALLY "1", THE ENTRY IN ITS PLACE IS A SCALE                        000280
C  FACTOR FOR THE INPUT. THIS DOES NOT AFFECT THE                       000290
C  OUTPUT.                                                              000300
C                                                                       000310
      DATA (AT(1,I),I=1,4)/-3.354,-12.24,0.,0./                         000320
      DATA (AT(2,I),I=1,4)/1.,-2.034,2*0./                              000330
      DATA (AT(3,I),I=1,4)/1.,3*0./                                     000340
      DATA (AT(4,I),I=1,4)/0.,-1077.8,1077.8,0./                        000350
      DATA (BT(I),I=1,4)/34.17,.331,0.,0./                              000360
      DATA (CT(I),I=1,4)/3*0.,1./                                       000370
      DATA (X(I),I=1,4)/4*0./                                           000380
      DATA NT,N,NOR1,NOR2/30,4,4,28/                                    000390
      DATA IC,L,NOR3,KF,NOTS,NSM/0.5,7,7,20,3,4/                        000400
      DATA TAU,NTP,A,W/.3,50(1,100.,.5/                                 000410
      DATA IDYS(1)/2HSET(>) VS ACHIEVED H/                              000420
      DATA IDYS(3)/2HTC=.5 TAU=.3 NSM=4  /                              000430
      DATA IDYS(5)/2HL=7 A=153 W=.5,.6,.2/                              000440
      DATA IDYS(7)/2HA.C.=STSLEVEL  0=03 /                              000450
      DATA IDYS(9)/2OH   TIME IN SECONDS  /                             000460
      DATA IDYS(11)/2CHPROFILE ALTITUDE(FT)/                            000470
      DATA IDYS(13)/2CHFIG  . ACHIEVED AND DESIRED ALTITUDE AS A FN OF T000480
     1 /                                                                000490
```

VITA

David E. Chaffin was born on 30 May 1948 in Pontotoc, MS. He graduated (Cum Laude) from Memphis State University in August 1971 with a BSEE degree and an ROTC Commission (Distinguished Graduate). He completed pilot training at Moody AFB, GA, in November 1973 and was assigned to Williams AFB, AZ, as an instructor pilot in the 96 FTS and subsequently as an 82 FTW flight examiner in the T-37. His assignment prior to entering AFIT was to the 41 MAS, Charleston AFB, SC, in the C-141.

Permanent address: 1871 Tahiti Lane
Memphis, TN 38117

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/GE/EE/79-9	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE *(and Subtitle)* THE APPLICATION OF OUTPUT PREDICTIVE DIGITAL CONTROL TO WING FLUTTER SUPPRESSION AND TERRAIN FOLLOWING PROBLEMS		5. TYPE OF REPORT & PERIOD COVERED MS Thesis 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) David E. Chaffin Captain		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Institute of Technology (AFIT-EN) Wright-Patterson AFB, Ohio 45433		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Flight Dynamics Laboratory (AFFDL-FGC) Air Force Systems Command Wright-Patterson AFB, Ohio 45433		12. REPORT DATE December, 1979 13. NUMBER OF PAGES 162
14. MONITORING AGENCY NAME & ADDRESS*(if different from Controlling Office)*		15. SECURITY CLASS. *(of this report)* Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT *(of this Report)*

Approved for public release; distribution unlimited

17. DISTRIBUTION STATEMENT *(of the abstract entered in Block 20, if different from Report)*

18. SUPPLEMENTARY NOTES

Approved for public release; IAW AFR 190-17

JOSEPH P. HIPPS, Major, USAF
Director of Information

19. KEY WORDS *(Continue on reverse side if necessary and identify by block number)*

Output Predictive Control
Regulator
Tracking
Terrain Following
Pitch Axis Autopilot
Robustness

20. ABSTRACT *(Continue on reverse side if necessary and identify by block number)*

This thesis is a study of a digital control technique known as Output Predictive Control (OPC) or Model Algorithmic Control (MAC). In OPC, the behavior of the system is predicted using its impulse response function and the desired response is characterized by a reference trajectory. Controls are computed iteratively to drive the system output along the desired trajectory. In an earlier study, the system was made to follow the reference trajectory exactly, but only at the control application time; there were large oscillations of the

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE *(When Data Entered)*

output between control changes. In this study, the control calculation is reformulated as a least squares curve fit problem, allowing some deviation from the desired trajectory. This approach is applied as a regulator for a very lightly damped fourth-order single-input/single-output system and as a pitch axis autopilot in a simplified terrain following problem. A qualitative discussion of robustness properties is included. The design of the controller is difficult due to the interrelationships of the internal parameters; however, the results of the terrain following example indicate that this is a viable approach for this problem.